



**TUGAS AKHIR - SS141501**

# **SUPPORT VECTOR MACHINE UNTUK DETEKSI KEJANG EPILEPSI BERDASARKAN SINYAL EEG**

**BAYU SAMUDRA  
NRP 062114 4000 0115**

**Dosen Pembimbing  
Dr. Santi Wulan Purnami, M.Si.**

**PROGRAM STUDI SARJANA  
DEPARTEMEN STATISTIKA  
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2018**





**TUGAS AKHIR - SS141501**

# **SUPPORT VECTOR MACHINE UNTUK DETEKSI KEJANG EPILEPSI BERDASARKAN SINYAL EEG**

**BAYU SAMUDRA  
NRP 062114 4000 0115**

**Dosen Pembimbing  
Dr. Santi Wulan Purnami, M.Si.**

**PROGRAM STUDI SARJANA  
DEPARTEMEN STATISTIKA  
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2018**





**FINAL PROJECT - SS 141501**

# **SUPPORT VECTOR MACHINE FOR EPILEPTIC SEIZURES DETECTION BASED ON EEG SIGNAL**

**BAYU SAMUDRA  
SN 062114 4000 0115**

**Supervisor  
Dr. Santi Wulan Purnami, M.Si.**

**UNDERGRADUATE PROGRAMME  
DEPARTMENT OF STATISTICS  
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2018**



## LEMBAR PENGESAHAN

### **SUPPORT VECTOR MACHINE UNTUK DETEKSI KEJANG EPILEPSI BERDASARKAN SINYAL EEG**


#### **TUGAS AKHIR**

Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Sains  
pada  
Program Studi Sarjana Departemen Statistika  
Fakultas Matematika, Komputasi, dan Sains Data  
Institut Teknologi Sepuluh Nopember

Oleh :

**Bayu Samudra**  
NRP. 06214 4000 0115

Disetujui oleh Pembimbing :  
**Dr. Santi Wulan Purnami, M.Si.**  
NIP. 19720923 199803 2 001

(  )



Mengetahui,  
Kepala Departemen

  
**Dr. Suhartono**

NIP. 19710929 199512 1 001

**SURABAYA, JULI 2018**

*(Halaman ini sengaja dikosongkan)*



# **SUPPORT VECTOR MACHINE UNTUK DETEKSI KEJANG EPILEPSI BERDASARKAN SINYAL EEG**

**Nama Mahasiswa** : Bayu Samudra  
**NRP** : 06214 4000 0115  
**Departemen** : Statistika-FMKSD ITS  
**Dosen Pembimbing** : Dr. Santi Wulan Purnami, M.Si.

## **Abstrak**

*Gangguan saraf yang paling umum ditemui salah satunya adalah epilepsi. Epilepsi merupakan penyakit yang dapat menyerang siapapun, tanpa melihat jenis kelamin ataupun usia. Diagnosis yang tepat dan penanganan yang tepat waktu terhadap epilepsi merupakan hal yang penting bagi pasien. Analisis EEG adalah metode yang umum dilakukan untuk mendiagnosa epilepsi, tetapi inspeksi visual oleh ahli saraf membutuhkan waktu yang relatif lama. Penelitian ini akan dilakukan deteksi kejang epilepsi dengan metode machine learning berdasarkan sinyal EEG. Data EEG yang digunakan pada tugas akhir ini berasal dari database yang tersedia di website milik Klinik für Epileptologie, Universität Bonn. Metode Discrete wavelet transform digunakan untuk mendekomposisi sinyal menjadi pita-pita frekuensi. Pita-pita frekuensi tersebut selanjutnya diekstraksi fitur energy, variance, entropy dan line length. Setelah didapatkan fitur-fitur pada setiap pita frekuensi selanjutnya dilakukan klasifikasi dengan metode Support vector machine (SVM) menggunakan kernel linear dan RBF. Hasil analisa menunjukkan bahwa metode SVM menggunakan kernel linear dan rbf memiliki hasil akurasi, spesitivitas, sensitivitas dan AUC masing-masing sebesar 98,4%, 98,75%, 97% dan 97,875%.*

**Kata Kunci:** *Accuracy, AUC, Discrete Wavelet Transform, Electroencephalographic, Epilepsi, Sensitivity, Specitify.*

*(Halaman ini sengaja dikosongkan)*

## **SUPPORT VECTOR MACHINE FOR EPILEPTIC SIEZURES DETECTION BASED ON EEG SIGNAL**

**Name** : Bayu Samudra  
**SN** : 06214 4000 0115  
**Department** : Statistics-FMKSD ITS  
**Supervisor** : Dr. Santi Wulan Purnami, M.Si.

### **Abstract**

*The most common neurological disorders encountered worldwide are epilepsy. Epilepsy is a disease that can attack anyone, regardless of gender or age. Two of the three patients were able to control the seizures satisfactorily with anti-seizure medication. EEG analysis is a common method for diagnosing epilepsy. The data used in this thesis comes from the database available on Klinik für Epileptologie, Universität Bonn website. The data amounts to 500 consisted of five sets of epilepsy seizures (set S) and not epileptic seizures (sets Z, N, O, F). Discrete wavelet transform method is used to decompose the signal into frequency bands. The bands are then extracted into features of energy, variance, entropy and line length. After obtaining the features of each frequency band, the following will be classified using the Support vector machine (SVM) method using linear kernel and rbf. The result of analysis shows that SVM method using linear kernel and RBF has accuracy, specificity, sensitivity and AUC of 98.4%, 98.75%, 97% and 97,875% respectively.*

**Keywords:** Accuracy, AUC, Discrete Wavelet Transform, Electroencephalographic, Epilepsi, Sensitivity, Specitify.

*(Halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

Mahasuci Allah ‘azza wa jalla tiada pengetahuan kami selain yang Dia ajarkan. Dialah tumpuhan harapan atas segala sesuatu atas kuasa-Nya penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“Support Vector Machine untuk Mendeteksi Kejang Epilepsi Berdasarkan Sinyal EEG”**.

Penulis menyadari Tugas Akhir ini terselesaikan sebab sampainya nikmat Allah berupa bantuan, bimbingan dan motivasi melalui berbagai pihak sebagai perantara. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Kedua orang tua yang telah memberi kasih sayang dalam membesarkan penulis.
2. Seluruh dosen yang telah membimbing dan membagi ilmu kepada penulis semasa perkuliahan.
3. Teman-teman sekontrakan, Bayu Aksan, Ardi Tandioga, Yoga Perdana Karra, Faisalmahdy, Noer Fajar, yang telah memberi bantuan selama masa awal kuliah di Surabaya
4. Andi Batari Ahmad yang telah menemani penulis melewati masa penulisan Tugas Akhir.
5. Teman-teman Statistika ITS angkatan 2014, Respect, yang selalu memberikan dukungan dan waktu kepada penulis selama masa perkuliahan.
6. Semua pihak yang turut membantu dalam pelaksanaan Tugas Akhir yang tidak bisa penulis sebutkan satu persatu.

Besar harapan penulis untuk mendapatkan kritik dan saran yang membangun sehingga Tugas Akhir ini dapat memberikan manfaat bagi semua pihak yang terkait.

Surabaya, Juli 2018

Penulis

*(Halaman ini sengaja dikosongkan)*

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>COVER PAGE .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN.....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>KATA PENGANTAR .....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xv</b>
<b>DAFTAR TABEL.....</b>	<b>xvii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xix</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan.....	4
1.4 Manfaat.....	4
1.5 Batasan Masalah .....	4
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1 <i>Electroencephalogram (EEG)</i> .....	5
2.2 <i>Discrete Walvet Transform (DWT)</i> .....	6
2.3 Ekstraksi Fitur .....	9
2.4 <i>Support Vector Machine (SVM)</i> .....	10
2.4.1 <i>Support Vector Machine Linier</i> <i>Separable</i> .....	11
2.4.2 <i>Support Vector Machine Linier</i> <i>Separation for nonseparable data</i> .....	12
2.4.3 <i>Support Vector Machine Non-Linier</i> <i>Separation for separable data</i> .....	14
2.4 Evaluasi Kinerja Klasifikasi dan <i>Stratified K-</i> <i>Fold Cross-Validation</i> .....	15
<b>BAB III METODOLOGI PENELITIAN.....</b>	<b>17</b>
3.1 Sumber Data .....	17
3.2 Sutruktur Data .....	17

3.4 Langkah Analisis .....	20
<b>BAB IV ANALISIS DAN PEMBAHASAN</b> .....	23
4.1 Eksplorasi Data Sinyal EEG .....	23
4.2 Dekomposisi Sinyal Menggunakan DWT .....	25
4.2.1 DWT Level 4.....	26
4.2.2 DWT Level 5.....	27
4.2.3 DWT Level 6.....	28
4.3 Ekstraksi Fitur.....	29
4.3.1 <i>Energy</i> .....	30
4.3.2 <i>Variance</i> .....	32
4.3.3 <i>Entropy</i> .....	34
4.3.4 <i>Line length</i> .....	36
4.4 <i>Support Vector Machine (SVM)</i> .....	39
4.4.1 <i>SVM Kernel Linear</i> .....	39
4.4.2 <i>SVM Kernel RBF</i> .....	43
<b>BAB V KESIMPULAN DAN SARAN</b> .....	49
5.1 Kesimpulan .....	49
5.2 Saran .....	49
DAFTAR PUSTAKA.....	51
LAMPIRAN .....	55
BIODATA PENULIS .....	85



## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b> Proses Dekomposisi Sinyal Sumber: .....	9
<b>Gambar 2.2</b> Ilustrasi SVM .....	11
<b>Gambar 3.1</b> Diagram Alir Langkah Analisis Secara Umum.....	22
<b>Gambar 4.1</b> Perbandingan Kategori Data .....	23
<b>Gambar 4.2</b> Plot Amplitudo Data Terhadap Waktu (a) Data Z, (b) Data O, (c) N, (d) F dan (e) S .....	24
<b>Gambar 4.3</b> Plot Gabungan 5 Set Data .....	25
<b>Gambar 4.4</b> <i>Boxplot</i> Level 4 .....	26
<b>Gambar 4.5</b> <i>Boxplot</i> Level 5 .....	28
<b>Gambar 4.6</b> <i>Boxplot</i> Level 6 .....	29
<b>Gambar 4.7</b> Plot Fitur <i>Energy</i> Kelima Set Data pada Tiap Sub-Sinyal (a) Sinyal A4, (b) Sinyal D4 dan (c) Sinyal D3, (d) Sinyal D2 dan (e) Sinyal D1 .....	31
<b>Gambar 4.8</b> Plot Fitur <i>Variance</i> Kelima Set Data pada tiap Sub-Sinyal (a) Sinyal A4, (b) Sinyal D4 dan (c) Sinyal D3, (d) Sinyal D2 dan (e) Sinyal D1 .....	33
<b>Gambar 4.9</b> Plot Fitur <i>Entropy</i> Kelima Set Data pada tiap Sub-Sinyal (A) Sinyal A4, (B) Sinyal D4, (C) Sinyal D3, (D) Sinyal D2 dan (E) Sinyal D1 .....	35
<b>Gambar 4.10</b> Plot Fitur <i>Line length</i> Kelima Set Data pada tiap Sub-Sinyal (A) Sinyal A4, (B) Sinyal D4, (C) Sinyal D3, (D) Sinyal D2 dan (E) Sinyal D1 .....	38

*(Halaman ini sengaja dikosongkan)*

## DAFTAR TABEL

	Halaman
<b>Tabel 2.1</b> <i>Confusion Matrix</i> .....	15
<b>Tabel 3.1</b> Struktur Data Sinyal Awal.....	18
<b>Tabel 3.2</b> Jumlah Sub-sinyal Tiap Level .....	18
<b>Tabel 3.3</b> Variabel Tahapan Klasifikasi DWT Level 4 .....	19
<b>Tabel 3.4</b> Struktur Data Tahapan Klasifikasi.....	20
<b>Tabel 4.1</b> Sub-Sinyal Level 4 .....	26
<b>Tabel 4.2</b> Sub-Sinyal Level 5 .....	27
<b>Tabel 4.3</b> Sub-Sinyal Level 6 .....	28
<b>Tabel 4.4</b> Hasil Ekstraksi Fitur <i>Energy</i> .....	30
<b>Tabel 4.5</b> Hasil Ekstraksi Fitur <i>Variance</i> .....	32
<b>Tabel 4.6</b> Hasil Ekstraksi Fitur <i>Entropy</i> .....	34
<b>Tabel 4.7</b> Hasil Ekstraksi Fitur <i>Line length</i> .....	37
<b>Tabel 4.8</b> Performa tiap <i>Fold SVM Linear</i> DWT Level 4 ....	40
<b>Tabel 4.9</b> <i>Confusion Matrix</i> level 4 .....	41
<b>Tabel 4.10</b> Performa tiap <i>Fold SVM Linear</i> DWT Level 5 ....	41
<b>Tabel 4.11</b> <i>Confusion Matrix</i> Level 5.....	42
<b>Tabel 4.12</b> Hasil Performa SVM <i>Linear</i> tiap <i>Fold</i> DWT Level 6 .....	42
<b>Tabel 4.13</b> <i>Confusion Matrix</i> Level 6.....	43
<b>Tabel 4.14</b> Hasil Performan SVM <i>Kernel RBF</i> tiap <i>Fold</i> pada DWT Level 4.....	44
<b>Tabel 4.15</b> <i>Confusion Matrix</i> Level 4.....	45
<b>Tabel 4.16</b> Hasil Performa SVM <i>Kernel RBF</i> pada DWT Level 5 .....	46
<b>Tabel 4.17</b> <i>Confusion Matrix</i> Level 5.....	46
<b>Tabel 4.18</b> Hasil Performa SVM <i>Kernel RBF</i> pada DWT Level 6 .....	47
<b>Tabel 4.19</b> <i>Confusion Matrix</i> Level 6.....	47

*(Halaman ini sengaja dikosongkan)*

## DAFTAR LAMPIRAN

	Halaman
<b>Lampiran 1.</b> Set Data EEG .....	55
<b>Lampiran 2.</b> <i>Syntax</i> DWT Level 4.....	56
<b>Lampiran 3.</b> <i>Syntax</i> DWT Level 5.....	58
<b>Lampiran 4.</b> <i>Syntax</i> DWT Level 6.....	60
<b>Lampiran 5.</b> Hasil DWT Level 4 .....	62
<b>Lampiran 6.</b> Hasil DWT Level 5 .....	63
<b>Lampiran 7.</b> Hasil DWT Level 6 .....	64
<b>Lampiran 8.</b> <i>Syntax</i> Fitur <i>Energy</i> Level 4.....	65
<b>Lampiran 9.</b> <i>Syntax</i> Fitur <i>Energy</i> Level 5.....	65
<b>Lampiran 10.</b> <i>Syntax</i> Fitur <i>Energy</i> Level 6.....	66
<b>Lampiran 11.</b> Data Fitur <i>Energy</i> Level 5.....	66
<b>Lampiran 12.</b> Data Fitur <i>Energy</i> Level 6.....	67
<b>Lampiran 13.</b> <i>Syntax</i> Fitur <i>Variance</i> Level 5 .....	67
<b>Lampiran 14.</b> <i>Syntax</i> Fitur <i>Variance</i> Level 6 .....	68
<b>Lampiran 15.</b> Data Fitur <i>Variance</i> Level 5.....	68
<b>Lampiran 16.</b> Data Fitur <i>Variance</i> Level 6.....	69
<b>Lampiran 17.</b> <i>Syntax</i> Fitur <i>Entropy</i> Level 5.....	69
<b>Lampiran 18.</b> <i>Syntax</i> Fitur <i>Entropy</i> Level 6.....	69
<b>Lampiran 19.</b> Data Fitur <i>Entropy</i> Level 5 .....	70
<b>Lampiran 20.</b> Data Fitur <i>Entropy</i> Level 6 .....	70
<b>Lampiran 21.</b> <i>Syntax</i> Fitur <i>Line length</i> Level 5.....	71
<b>Lampiran 22.</b> <i>Syntax</i> Fitur <i>Line length</i> Level 6.....	71
<b>Lampiran 23.</b> Data Fitur <i>Line length</i> Level 5 .....	72
<b>Lampiran 24.</b> Data Fitur <i>Line length</i> Level 6 .....	72
<b>Lampiran 25</b> <i>Syntax</i> SVM Kernel <i>Linear</i> .....	72
<b>Lampiran 26</b> <i>Syntax</i> SVM Kernel <i>RBF</i> .....	74
<b>Lampiran 27</b> <i>Confusion Matrix</i> Kernel <i>Linear</i> Dwt Level 4 per <i>Fold</i> .....	77

<b>Lampiran 28</b>	<i>Confusion Matrix Kernel RBF Dwt Level 4</i>	
	per Fold .....	78
<b>Lampiran 29</b>	<i>Confusion Matrix Kernel Linear Dwt Level</i>	
	5 per Fold .....	79
<b>Lampiran 30</b>	<i>Confusion Matrix Kernel RBF Dwt Level 5</i>	
	per Fold .....	80
<b>Lampiran 31</b>	<i>Confusion Matrix Kernel Linear Dwt Level</i>	
	5 per Fold .....	81
<b>Lampiran 32</b>	<i>Confusion Matrix Kernel RBF Dwt Level 5</i>	
	per Fold .....	82
<b>Lampiran 33</b>	Model SVM Kernel RBF pada DWT level 4 .....	83
<b>Lampiran 34</b>	Surat Pernyataan Data .....	84

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Kelainan neurologis merupakan penyakit yang terjadi pada sistem saraf pusat dan *peripheral*. Ratusan juta orang di seluruh dunia menderita kelainan saraf (WHO, 2018). Terdapat lebih dari enam ratus penyakit yang merupakan kelainan saraf, seperti tumor otak, epilepsi, penyakit *parkinson's*, dan stroke serta penyakit yang kurang familial seperti *frontotemporal dementia* (Francisco, 2018). Salah satu kelainan saraf yang paling umum dijumpai adalah epilepsi. Menurut World Health Organization (WHO), sekitar lima puluh juta orang menderita epilepsi saat ini. Diperkirakan 2,4 juta orang terdiagnosis epilepsi setiap tahunnya (WHO, 2018). Epilepsi adalah gangguan saraf kronik yang ditandai dengan kejang berulang. Kelainan saraf ini merupakan penyakit yang dapat menyerang siapapun, tanpa melihat jenis kelamin ataupun usia. Epilepsi dapat terjadi disebabkan beberapa faktor seperti demam, kurang tidur, kelainan genetik, tumor, trauma, keracunan, hipoksia, inflamasi, dan pembengkakan sel (Silbernagl & Lang, 2000). Epilepsi adalah penyakit yang secara medis tidak dapat disembuhkan. Obat yang dikonsumsi penderita epilepsi dimaksudkan untuk mengontrol frekuensi terjadinya kejang.

Epilepsi adalah kelainan saraf yang ditandai dengan kejang berulang. Seseorang dapat dikatakan menderita epilepsi bila sudah pernah mengalami kejang lebih dari sekali. Orang yang mengalami kejang bukan berarti penderita epilepsi. Kejang disebabkan oleh pelepasan muatan elektrik yang tidak terkontrol dalam sekelompok neuro (Acharya, dkk., 2017). Sel-sel saraf saling berkomunikasi dengan mengirimkan sinyal listrik dalam pola yang teratur. Pada penderita epilepsi sinyal listrik ini beraktivitas melebihi batasan normal. Sinyal yang tidak normal ini kemudian menyebar ke area otak disekitarnya dan menciptakan lonjakan sinyal listrik yang tidak terkendali. Sinyal listrik ini akan disampaikan ke otot yang akan menyebabkan kejang. Gerakan tubuh yang dilakukan tanpa

sengaja, serta ketidaksadaran merupakan gejala umum yang diamati selama masa kejang. Kejang umumnya kambuh dan bertahan dalam jangka waktu tertentu. Kejang non-epilepsi dapat disebabkan oleh stres atau masalah psikologis. Pemeriksaan elektroensefalogram (EEG) dan respon terhadap obat anti epilepsi merupakan contoh petunjuk yang dapat membedakan kejang epilepsi dan non-epilepsi.

Diagnosis epilepsi saat ini dilakukan dengan mengumpulkan riwayat terperinci, melakukan pemeriksaan neurologis dan pemeriksaan tambahan seperti *neuro-imaging* dan *electroencephalographic* (EEG). EEG adalah salah satu tes yang dilakukan untuk mengukur aktivitas kelistrikan dari otak untuk mendeteksi adanya kelainan dari otak. Pemeriksaan ini menggunakan sensor khusus berupa elektroda yang dipasang di kepala dan dihubungkan melalui kabel menuju komputer. EEG akan merekam aktivitas listrik dari otak yang direpresentasikan dalam bentuk gelombang. Sinyal ini menyediakan informasi penting tentang jaringan epileptogenik yang harus dianalisis dan dipahami sebelum dilakukan inisiasi proses terapi (Faust, dkk., 2015). Diagnosis menggunakan EEG dapat dilakukan untuk mendeteksi penyakit epilepsi. Hasil pemeriksaan EEG memberikan gambaran aktivitas otak yang menunjukkan peningkatan resiko terjadinya serangan epilepsi. Umumnya, dokter mengambil hasil dari rekaman EEG dan mengidentifikasi apakah terdapat tanda epilepsi atau tidak (Sandeep, dkk., 2017). Bagaimanapun, pemeriksaan rekaman EEG secara inspeksi visual oleh seorang ahli saraf membutuhkan waktu yang lama. Oleh sebab itu, diperlukan sistem *computer-aided diagnosis (CAD)* untuk mendapatkan diagnosis yang akurat, cepat dan objektif.

Penelitian mengenai sinyal EEG telah banyak dilakukan dengan berbagai metode yang berbeda-beda dari metode *features extraction* maupun metode klasifikasinya. Diantaranya, penelitian oleh (Kannathal , dkk., 2005) yang menggunakan *entropy estimator* dalam membedakan data EEG normal dan epilepsi dengan menggunakan *adaptive neuro-fuzzy inference system*



(ANFIS) *classifier*. Dari hasil penelitian tersebut menunjukkan bahwa estimator entropi dapat membedakan data EEG normal dan epilepsi. Penelitian lain menggunakan *discrete wavelet transform* (DWT) dan *support vector machine* (SVM). DWT mendekomposisi sinyal EEG menjadi kelompok frekuensi. Kelompok frekuensi tersebut diekstraksi sehingga mendapatkan fitur statistik seperti *energy*, *variance* dan *zero crossing rate* serta fitur nonlinier seperti *fractal dimension*. SVM *classifier* digunakan untuk mengklasifikasikan sinyal EEG kejang dan tidak kejang (Kumar & Kolekar, 2014). Penelitian lain oleh (Acharya, dkk., 2011) mengklasifikasikan rekaman sinyal EEG menjadi normal, ictal, dan interictal menggunakan *recurrence plots* (RP) dan ekstraksi *recurrence quantification analysis* (RQA). Fitur yang diperoleh dengan RQA kemudian dilanjutkan dengan menggunakan tujuh metode klasifikasi yang berbeda yaitu SVM, *gaussian mixture model*, *fuzzy sugeno classifier*, *K-nearest neighbor*, *naive bayes classifier*, *decision tree* dan *radial basis probabilistic neural network*. Hasil penelitian tersebut menunjukkan SVM dapat mengidentifikasi kelas EEG dengan rata-rata *efficiency* 95,6%, *sensitivity* dan *specificity* sebesar 98,9% dan 97,8%. Penelitian lain menggunakan algoritma *permutation entropy* sebagai *features extraction* dan SVM sebagai metode klasifikasi yang menghasilkan akurasi sebesar 86,10% (Tzallas, dkk., 2007).

Penelitian sebelumnya mengenai klasifikasi sinyal EEG menunjukkan DWT menjadi metode yang paling sesuai. Sebab, DWT memiliki kelebihan dari domain *time-frequency* serta lebih sesuai untuk analisi sinyal non stasioner (Mirzaei, dkk., 2010). Pada penelitian *Application of Recurrence Quantification Analysis for The Automated Identification of Epileptic EEG Signal* setelah melakukan eksperimen dengan beberapa *classifiers* didapatkan SVM *classifier* menghasilkan akurasi sebesar 95,6% (Acharya, dkk., 2011).

Pada penelitian ini akan dilakukan klasifikasi kejang dan non berdasarkan sinyal EEG menggunakan metode klasifikasi

*Support Vector Machine* (SVM). Sebelum dilakukan klasifikasi pasien menggunakan SVM *classifier* terlebih dahulu dilakukan dekomposisi sinyal menjadi beberapa sub-sinyal dengan metode DWT. Metode ini digunakan untuk memberikan penilaian kuantitatif dari pita frekuensi berbeda.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka terdapat beberapa permasalahan yang akan dibahas dalam penelitian ini. Permasalahan pertama adalah bagaimana merepresentasikan sinyal EEG menjadi berdomain *time frequency*. Permasalahan kedua adalah mengenai seberapa besar akurasi, sensitivitas, spesifitas dan AUC dari hasil klasifikasi sinyal EEG.

## 1.3 Tujuan

Berdasarkan rumusan masalah tersebut, tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut:

1. Mendapatkan pita-pita frekuensi dari data sinyal EEG dan ekstraksi fitur dari masing-masing pita frekuensi.
2. Mengetahui akurasi, sensitivitas dan spesifitas dari hasil klasifikasi berdasarkan data sinyal EEG menggunakan SVM.

## 1.4 Manfaat

Analisis ini diharapkan dapat memberikan tambahan informasi dalam menggunakan metode yang tepat untuk pengklasifikasian pasien epilepsi sehingga membantu mempercepat proses diagnosis pasien dengan tepat.

## 1.5 Batasan Masalah

Data yang digunakan merupakan data EEG pasien yang diunduh dari website milik "Klinik für Epileptologie, Universität Bonn".

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini membahas mengenai *Electroencephalogram*, DWT, ekstraksi fitur, *Cross validation*, *Support Vector Machine*, serta evaluasi kinerja klasifikasi.

#### **2.1 *Electroencephalogram* (EEG)**

Aktivitas listrik dapat dideteksi di sepanjang kulit kepala sebagai hasil dari arus ion yang mengalir didalam sekelompok neuron di otak. Aktivitas ini dapat direkam dalam bentuk *Electroencephalogram* (EEG). Elektroda dipasang disepanjang kulit kepala untuk merekam aktivitas otak pada posisi yang berbeda. EEG mengandung informasi penting terkait keadaan otak. EEG memegang peranan penting dalam penelitian otak terutama pada diagnosis dan klasifikasi penyakit saraf seperti epilepsi, schizophrenia dan Parkinson. Rekaman EEG kejang umumnya mengandung amplitudo tinggi yang abnormal. Objek yang direkam adalah aktivitas bioelektrikal neuron-neuron pada lapisan cortex cerebrum. Hasil rekaman yang didapatkan melalui metode ini berupa sinyal gelombang otak, yang berada pada range tegangan 5-100 mV.

Sinyal EEG ditangkap melalui sensor pada elektroda yang dapat terpasang baik secara *invasive* maupun *non invasive*. Pemasangan secara *invasive* memerlukan adanya implant elektroda-elektroda pada bagian-bagian otak yang ingin diamati responnya secara langsung. Metode ini menghasilkan sinyal yang lebih bersih dari *noise* spasial, karena sinyal yang didapat berasal dari bagian otak yang terkait secara langsung. Sedangkan, metode pemasangan secara *non invasive*, *channel* elektroda diletakkan di sekeliling tempurung kepala, sehingga memungkinkan sinyal gelombang otak tetap diperoleh. Kelemahan metode ini ada pada sinyal yang dihasilkan, dimana selalu terdapat inferensi antara hasil perekaman satu *channel* dengan *channel* lainnya, tetapi metode ini lebih aman dan mudah.

## 2.2 Discrete Walvet Transform (DWT)

Transformasi adalah proses merepresentasikan suatu sinyal ke dalam domain atau kawasan lain. Tujuan dari transformasi adalah untuk lebih menonjolkan sifat atau karakteristik sinyal tersebut. Definisi wavelet (secara harfiah berarti “gelombang kecil”) adalah himpunan fungsi dalam ruang vektor  $L^2\mathbf{I}$ , yang mempunyai sifat-sifat sebagai berikut (Burrus, dkk., 1998):

- berenergi terbatas
- merupakan fungsi band-pass pada domain frekuensi
- merupakan hasil penggeseran (translasi) dan penskala (dilatasi) dari
- sebuah fungsi tunggal (induk), yaitu

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \quad (2.1)$$

Dengan  $a, b \in R$  (bilangan nyata), dan  $a \neq 0$ .  $a$  adalah parameter penskalaan dan  $b$  adalah parameter penggeser posisi terhadap sumbu  $t$ . Faktor normalisasi  $|a|^{-\frac{1}{2}}$  digunakan untuk memastikan bahwa  $\|\psi_{a,b}(t)\| = \|\psi(t)\|$ .

Teori dasar dari transformasi *wavelet* didasari oleh pembangkitan sejumlah filter dengan cara menggeser dan menskala suatu *mother wavelet* berupa *band-pass filter*. Dengan demikian hanya diperlukan pembangkitan sebuah filter. Filter lain mengikuti aturan penskalaan, baik pada kawasan waktu maupun kawasan frekuensi. Penambahan skala wavelet akan meningkatkan durasi waktu, mengurangi lebar bidang dan menggeser frekuensi pusat ke nilai frekuensi yang lebih rendah. Sebaliknya pengurangan skala menurunkan durasi waktu, menambah lebar bidang dan menggeser frekuensi ke nilai frekuensi yang lebih tinggi (Burrus, dkk., 1998). Perapatan ( $0 < a < 1$ ) dan peregangan ( $a > 1$ ) akan menskala tanggapan frekuensi wavelet yang dibangkitkan, sehingga menghasilkan sejumlah wavelet yang

mencakup rentang frekuensi yang diinginkan. Kumpulan wavelet ini dapat dianggap sebagai suatu *filter bank* untuk analisis sinyal.

Keuntungan Transformasi *Wavelet* adalah bahwa jendelanya bervariasi. Untuk mengisolasi ketidakkontinuan sinyal, dapat digunakan fungsi basis yang sangat pendek. Pada saat yang sama, untuk analisis frekuensi secara terperinci, dapat digunakan fungsi basis yang sangat panjang.

Transformasi *Wavelet* dibedakan menjadi 2, yaitu Transformasi *Wavelet* Kontinu (*Continuous Wavelet Transform/CWT*) dan Transformasi *Wavelet* versi diskrit. Versi diskrit dikenal dengan *Discret Wavelet Transform* (DWT). Pada Transformasi *Wavelet* Kontinu, waktu  $t$  serta parameter penskala  $a$  dan penggeser  $b$  berubah secara kontinu (dengan  $a \neq 0$ ). Transformasi *Wavelet* kontinu fungsi  $f(t)$  didefinisikan dengan :

$$\begin{aligned} CWT &= \langle \Psi_{a,b}(t), f(t) \rangle \\ &= \int_{-\infty}^{\infty} \Psi_{a,b}(t) f(t) dt \\ &= \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \Psi\left(\frac{t-b}{a}\right) f(t) dt \end{aligned} \quad (2.2)$$

Transformasi *Wavelet* Kontinu ini mempunyai dua kelemahan, yaitu redundancy dan ketidakpraktisan (Mallat, 1999). Masalah tersebut dapat diselesaikan dengan mendiskritkan parameter  $a$  dan  $b$ . Transformasi yang bersifat diskrit, dilakukan pendiskritan terhadap parameter  $a$  dan  $b$ , dengan  $a = a_0^j$  dan  $b = a_0^j k b_0$  dimana  $j$  dan  $k$  bilangan bulat, serta  $a_0 > 1$  dan  $b_0 > 0$  dan membuat waktu  $t$  menjadi diskrit. Pemilihan nilai  $a_0$  dan  $b_0$  bergantung pada wavelet  $\psi$  yang maka akan diperoleh DWT dengan persamaan sebagai berikut:

$$TWD(a_0^j, a_0^j k b_0) = \frac{1}{\sqrt{a_0^j}} \sum_n \psi\left(\frac{n}{a_0^j} - k b_0\right) f(n) \quad (2.3)$$

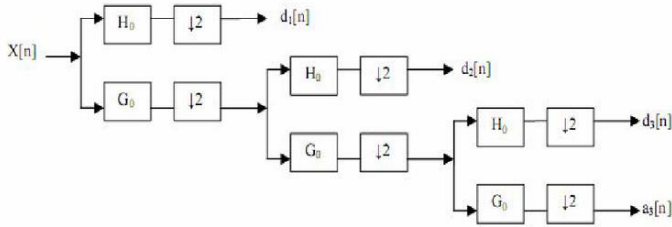
Dengan  $\psi_{m,n}(t) = \psi^{a_0^m n b_0 a_0^m}(t) = a_0^{-\frac{m}{2}} \psi(a_0^{-m} t - n b_0)$ .

Proses dekomposisi sinyal dilakukan menggunakan metode *subband coding*. *Subband coding* adalah proses dekomposisi sinyal menjadi frekuensi yang berbeda diperoleh dari proses filter. Proses filter DWT melewati sinyal masukan melalui *high pass filter*  $g[i]$  dan *low pass filter*  $h[i]$  dengan persamaan sebagai berikut:

$$a(k) = \sum_i^n x(i) h(-i + 2k) \quad (2.4)$$

$$d(k) = \sum_i^n x(i) g(-i + 2k) \quad (2.5)$$

Dengan  $a(k)$  dan  $d(k)$  adalah *output* dari *high pass filter* dan *low pass filter*. Setelah proses *filtering* separuh dari sampel sampel dieliminasi atau *downsampling* berdasarkan aturan *Nyquist* (Kumar & Kolekar, 2014). Proses *filtering* dimana  $x(i)$  adalah fungsi sinyal yang menjadi *input filtering*, fungsi  $g[-i + 2k]$  dan  $h[-i + 2k]$  pada Persamaan 2.4 dan 2.5 adalah fungsi koefisien wavelet untuk *high pass* dan *low pass*  $i$  adalah panjang sinyal *input*,  $k$  adalah panjang koefisien wavelet yang digunakan. *Output* dari *high pass filter* adalah detail (D) dan *output* dari *low pass filter* adalah aproksimasi (A). dekomposisi satu level akan menghasilkan *output* D1 dan A1. Proses DWT level dua *output* A1 akan difilter lagi hingga menghasilkan *output* D2 dan A2. Proses tersebut berulang hingga N-level yang diinginkan. D1,D2,...,DN dan AN menjadi *output*.



**Gambar 2.1** Proses Dekomposisi Sinyal Sumber:  
“www.researchgate.net”

## 2.3 Ekstraksi Fitur

Setelah dilakukan dekomposisi sinyal EEG maka frekuensi tersebut diekstraksi sehingga mendapatkan fitur-fitur yang diinginkan berikut fitur yang dapat diekstrak (Hamad, dkk., 2016):

### 1. *Energy*

Energi dari sinyal didefinisikan sebagai jumlah kuadrat dari nilai modulus sampel. *Energy* dari sinyal dapat dirumuskan sebagai berikut:

$$E = \sum_{i=1}^n |X_i|^2 \quad (2.6)$$

Dimana  $X_i$  adalah nilai sampel, dan  $n$  adalah jumlah sampel masing-masing sub-band (Kumar & Kolekar, 2014).

### 2. *Variance*

Pengukuran *Variance* menunjukkan seberapa jauh persebaran dari sekumpulan data. *Variance* dari sekumpulan data dapat dihitung dengan menggunakan persamaan berikut.

$$V = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1} \quad (2.7)$$

Dengan  $\bar{X}$  adalah *mean* dari data sampel  $X$ , dan  $n$  adalah jumlah sampel (Kumar & Kolekar, 2014).

### 3. Entropy

*Entropy* merupakan pengukuran ketidakpastian secara numerik dari *outcome* letak sinyal yang mengandung ribuan bit informasi. Secara matematis *Entropy* dirumuskan dengan Persamaan

$$(EN) = \sum_{i=1}^{n-1} X_i^2 \log(X_i^2) \quad (2.8)$$

### 4. Line length

*Line length* adalah ukuran kompleksitas sinyal atau *waveform fractal dimention*. *Line length* sensitif terhadap amplitudo dan variasi frekuensi sinyal sehingga berlaku juga sebagai ukuran kekejangan atau patologi dari karakteristik kombinasi amplitudo dan frekuensi pada sinyal EEG (Esteller, dkk., 2001) dengan persamaan

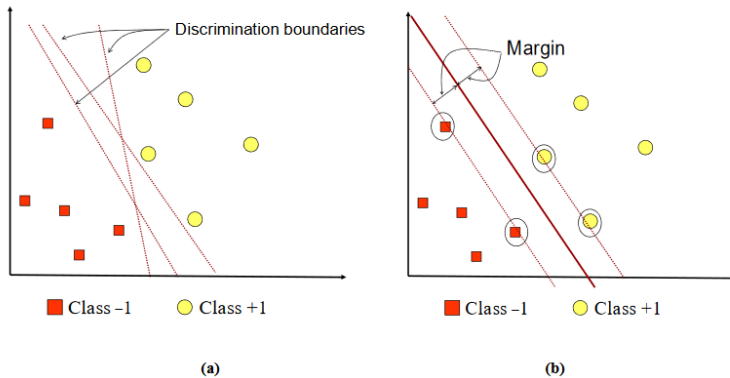
$$L = \frac{1}{n-1} \sum_{i=1}^{n-1} \text{abs}(x_{i+1} - x_i) \quad (2.9)$$

## 2.4 Support Vector Machine (SVM)

*Support Vector Machine* (SVM) merupakan suatu teknik yang relative baru dan saat ini telah banyak digunakan untuk melakukan klasifikasi yang sangat populer belakangan ini. SVM berada dalam satu kelas dengan *Neural Network* dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan, keduanya masuk kedalam kelas *supervised learning* (Santoso, 2007).

Secara teoritik SVM dikembangkan untuk menyelesaikan permasalahan klasifikasi pada dua kelas dengan mencari *Hyperplane* terbaik. *Hyperplane* merupakan fungsi untuk memisahkan antara dua kelas pada *input space*, sehingga dari data yang tersebar dapat dilakukan klasifikasi dan analisis regresi. Vapnik menjelaskan bahwa setiap permasalahan dapat dimodelkan menggunakan SVM (Vapnik, 2000). Ilustrasi SVM ditunjukkan pada Gambar 2.3





**Gambar 2.2** Ilustrasi SVM

Sumber: “*Support Vector Machine Teori dan Aplikasi dalam BioInformatika*”, 2003

Gambar 2.3 (a) *discrimination boundaries* ditunjukkan oleh beberapa garis dimana *class -1* disimbolkan dengan kotak (merah), sedangkan pada *class +1* disimbolkan dengan lingkaran (kuning). Gambar 2.3(b) menunjukkan bahwa terdapat garis *hyperplane* yang tepat diantara dua kelas. Prinsip dasar dari analisis ini adalah menemukan *hyperplane* terbaik yakni dengan meminimalkan kesalahan klasifikasi dan memaksimalkan margin geometriknya seperti pada Gambar 2.3(b). Usaha untuk menentukan *hyperplane* ini merupakan proses pembelajaran pada SVM (Nugroho, dkk., 2003).

#### 2.4.1 *Support Vector Machine Linier Separable*

Prinsip dasar SVM adalah *linier classifier*, yaitu kasus klasifikasi yang dapat dipisahkan secara linier. Data yang tersedia dinotasikan sebagai  $x_i \in R^d$  sedangkan label masing-masing dinotasikan  $y_i \in \{-1, +1\}$  untuk  $i = 1, 2, \dots, l$ , yang mana  $l$  adalah jumlah data. Diasumsikan kedua kelas -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane* berdimensi  $d$ , yang didefinisikan (Nugroho, dkk., 2003):

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0 \quad (2.10)$$

*Pattern*  $x_i$  yang termasuk *class* -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan dimana nilai menunjukkan adanya penurunan

$$\mathbf{w}^T \cdot \mathbf{x}_i + b \leq -1 \quad (2.11)$$

Sedangkan *pattern*  $x_i$  yang termasuk *class* +1 (sampel positif) menunjukkan adanya kenaikan pada nilai

$$\mathbf{w}^T \cdot \mathbf{x}_i + b \geq +1 \quad (2.12)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu  $1/\|\mathbf{w}\|$ . Hal ini dapat dirumuskan sebagai *quadratic programming* (QP) problem, yaitu mencari titik minimal Persamaan (2.13), dengan memerhatikan *constrain* Persamaan (2.14).

$$\min_{\mathbf{w}} \tau(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.13)$$

$$y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) - 1 \geq 1, i = 1, 2, \dots, n \quad (2.14)$$

*Problem* ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya *Lagrange Multiplier*

$$L_d = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j) \quad (2.15)$$

Persamaan  $L_d$  didapatkan dengan mensubstitusikan nilai  $y_{m_1}$ ,  $y_{m_2}$ ,  $\mathbf{x}_{m_1}$ , dan  $\mathbf{x}_{m_2}$  ke Persamaan (2.10). persamaan  $L_d$  digunakan untuk mencari nilai-nilai  $\alpha_m$  (*support vector*) dengan membuat  $L_d$  optimum.  $L_d$  optimum didapat dengan cara mencari turunan parsial  $L_d$  terhadap  $\alpha$ . Setelah mendapatkan nilai  $\alpha$ , langkah selanjutnya adalah mencari nilai  $\mathbf{w}$  dan  $b$  dengan persamaan sebagai berikut.

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \text{ dan } b = 1 - \mathbf{w}^T \cdot \mathbf{x} \quad (2.16)$$

#### 2.4.2 Support Vector Machine Linier Separation for nonseparable data

Beberapa kasus terkadang data tidak dapat dipisahkan dengan menggunakan pemisah linear. Namun, *hyperplane* dengan tingkat *error* yang paling minimum dapat dicari. Sehingga nantinya akan

ada variabel *slack* non negatif  $\xi_i, i = 1, \dots, m$ . Sehingga diperoleh persamaan dibawah ini:

$$(\mathbf{w}^T \cdot \mathbf{x}) + b \geq +1 - \xi_i \rightarrow y_i = +1 \quad (2.17)$$

$$(\mathbf{w}^T \cdot \mathbf{x}) + b \leq -1 - \xi_i \rightarrow y_i = -1 \quad (2.18)$$

Untuk menentukan *hyperplane* dan meminimasi *error* yang dihasilkan, fungsi tujuan dari permasalahannya adalah sebagai berikut:

$$\min_{w,b,\xi} \frac{1}{2} \|\mathbf{W}\|^2 + C \sum_{i=1}^n \xi_i \quad (2.19)$$

Batasan:

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) \geq 1 - \xi, \xi \geq 0, i = 1, \dots, p \quad (2.20)$$

$C$  adalah parameter yang ditentukan untuk mengontrol hubungan antara *slack* variabel dengan  $\|\mathbf{w}\|^2$ . Bentuk dual dari persamaan *Lagrange* adalah:

$$\max \mathbf{w}(x) = \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.21)$$

Batasan:

$$\arg \min \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^n \alpha_i \quad (2.22)$$

Dengan kendala pada *Lagrange Multiplier* persamaan dihasilkan persamaan untuk  $\mathbf{w}$  seperti persamaan berikut:

$$\mathbf{w} = \sum_{i=1}^n \alpha_j y_i \mathbf{x}_j \text{ dan } \hat{b} = -\frac{1}{2} \mathbf{w}(\mathbf{x}_r + \mathbf{x}_s) \quad (2.23)$$

Dimana  $\mathbf{x}_r$  dan  $\mathbf{x}_s$  adalah *support vector* dari tiap kelas, sehingga setelah mendapatkan nilai parameter tersebut, *hyperplane* dapat ditemukan

### 2.4.3 Support Vector Machine Non-Linear Separation for separable data

Pada kasus pemisahan non-linier digunakan fungsi *mapping* yang disebut dengan fungsi *kernel*. Fungsi ini dapat melakukan *mapping input space* dari data training ke *feature space* dimensi yang lebih tinggi. Fungsi *kernel* dapat dilihat pada persamaan berikut

$$(\phi(x_i)\phi(x_i)) = k(\mathbf{x}_i, \mathbf{x}) \quad (2.24)$$

Pada pemisahan nonlinier rumus *Lagrange* juga dapat digunakan untuk menemukan solusi.

$$\max W(\alpha) = \sum_{i=1}^p \alpha_i - \sum_{i=1}^p \sum_{j=1}^p \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}) \quad (2.25)$$

untuk melakukan prediksi dapat menggunakan Persamaan (2.34)

$$f(x) = \text{sign} \left( \sum_{i=1}^p y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \hat{\mathbf{b}} \right) \quad (2.26)$$

Dimana  $K(\mathbf{x}_i, \mathbf{x})$  merupakan matriks *kernel*. Fungsi pembentuk matriks *kernel* untuk menyelesaikan permasalahan *nonlinear* adalah sebagai berikut:

- a. *Kernel Gaussian* (RBF)

$$K(x_i x_j) = \exp \left( -\gamma \|x_i - x_j\|^2 \right) \quad (2.27)$$

- b. *Kernel linear*

$$K(x_i x_j) \quad (2.28)$$

## 2.4 Evaluasi Kinerja Klasifikasi dan *Stratified K-Fold Cross-Validation*

Sebelum dilakukan tahap klasifikasi data terlebih dahulu dilakukan data dibagi menjadi data *training* dan *testing*. *Stratified k-fold cross-validation* merupakan salah satu metode pembagian data *unbalance*. Pada *stratified k-fold cross-validation* data akan dipartisi secara acak menjadi  $k$  bagian atau *folds* dengan jumlah sampel sama yaitu  $D_1, D_2, \dots, D_k$  dan masing masing-masing *fold* memiliki proporsi yang sama antar kelas data. Validasi menggunakan *training* dan *testing* dilakukan sebanyak  $k$  kali. Pada iterasi ke- $i$ , partisi  $D_i$  akan diatur sebagai data *testing* dan partisi lainnya akan digunakan sebagai data *training*. Artinya, pada iterasi yang pertama, partisi  $D_2, D_3, \dots, D_k$  akan menjadi data *training* untuk mendapatkan model pertama yang akan diuji dengan partisi  $D_1$ . Pada iterasi kedua partisi  $D_1, D_3, \dots, D_k$  akan menjadi data *training* kemudian  $D_2$  akan menjadi data *testing*, begitu seterusnya sampai dengan  $D_k$  yang akan menjadi data *testing* (Han, dkk., 2012).

Setelah dilakukan pembagian data *training* dan *testing* akan dilakukan penghitungan performansi klasifikasi. Performansi klasifikasi dapat dihitung berdasarkan *confusion matrix* seperti pada Tabel 2.1.

**Tabel 2.1** *Confusion Matrix*

	Prediksi (+)	Prediksi (-)
Aktual (+)	True Positives (TP)	False Negatives (FN)
Aktual (-)	False Positives (FP)	True Negatives (TN)
	Total	

Kinerja klasifikasi bisa dievaluasi dengan memperhatikan beberapa ukuran sebagai berikut:

### 1. Akurasi

Akurasi adalah ukuran untuk mengukur ketepatan prediksi pengklasifikasian pada kelas tertentu. Rumus untuk menghitung nilai akurasi sesuai dengan persamaan (2.29)

$$Akurasi = \frac{\text{jumlah klasifikasi benar}}{\text{total data}} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.29)$$

## 2. *Sensitivity*

Nilai *sensitivity* adalah probabilitas dari hasil klasifikasi positif yang sesuai dari hasil klasifikasi yang memiliki kondisi target (Starlen, dkk., 2009) (2.30)

$$Sensitivity = \frac{TP}{TP+TN} \quad (2.30)$$

## 3. *Specificity*

Nilai *Specificity* adalah probabilitas dari hasil klasifikasi negatif yang sesuai terhadap aktual negative (Starlen, dkk., 2009) (2.31)

$$Specificity = \frac{TN}{TN+FP} \quad (2.31)$$

## 4. *Area Under ROC Curve (AUC)*

AUC adalah indikator ringkasan dari performansi kurva ROC yang dapat merangkum performansi tunggal. (Bekkar, dkk., 2013) mengusulkan untuk membuat pendekatan AUC dalam kasus *binary learning* dengan *balanced accuracy* yang ditunjukkan

$$AUC = 1/2(sensitivity + specificity) \quad (2.32)$$

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Sumber Data**

Data yang digunakan pada penelitian ini adalah data sekunder yang diperoleh dari database yang tersedia di website milik Klinik für Epileptologie, Universität Bonn. Data berupa hasil rekam sinyal EEG terhadap lima relawan dengan kondisi normal dan lima pasien penderita epilepsi. Data ini terdiri dari lima set rekaman EEG (Z,O,N,F,S) setiap set data berisi 100 data sinyal single-channel masing-masing data berdurasi 23,6 detik. Digitalisasi data dengan frekuensi 173,61 sampel per detik menggunakan A/D converter 12 bit. Masing-masing data sinyal EEG digital tersebut terdiri dari 4097 data diskrit. Kelima set data diambil dari orang yang berbeda beda dengan kondisi yang berbeda pula. Untuk set data Z dan O diambil dari relawan normal dengan kondisi yang berbeda. Set data O direkam dengan kondisi mata terbuka, sedangkan Z diambil dalam kondisi mata tertutup. Set data N, F dan S data sinyal otak dari pasien penderita epilepsi dimana set data N dan F pasien dalam keadaan tidak kejang tetapi set data S dalam kondisi kejang.

#### **3.2 Sutruktur Data**

Struktur data yang akan digunakan dalam tahap dekomposisi sinyal ditampilkan pada Tabel 3.1. nilai X pada Tabel 3.1 menunjukkan nilai amplitudo dari sinyal EEG. Amplitudo tersebut didapatkan dari beda potensial antara dua elektroda yang dipasang pada kulit pasien. Nilai X akan naik ketika salah satu bagian otak yang dipasang elektroda bermuatan negatif sedangkan pasangan elektrodanya bermuatan positif. Sedang jika terjadi sebaliknya nilai X akan turun dari nilai sebelumnya. Jika tidak terdapat perbedaan potensial antara kedua elektroda yang dipasang berpasangan pada kulit kepala pasien maka nilai X yang dihasilkan akan cenderung stabil.

**Tabel 3.1** Struktur Data Sinyal Awal

Set	Amplitudo Ke-				
		1	2	...	4097
F	1	X[1,1,1]	X[1,1,2]	...	X[1,1,4097]
	2	X[1,2,1]	X[1,2,2]	...	X[1,2,4097]
	:	:	:	:	:
	100	X[1,100,1]	X[1,100,2]	...	X[1,100,4097]
N	1	X[2,1,1]	X[2,1,2]	...	X[2,1,4097]
	2	X[2,2,1]	X[2,2,2]	...	X[2,2,4097]
	:	:	:	:	:
	100	X[2,100,1]	X[2,100,2]	...	X[2,100,4097]
:	:	:	:	:	:
Z	1	X[5,1,1]	X[5,1,2]	...	X[5,1,4097]
	2	X[5,2,1]	X[5,2,2]	...	X[5,2,4097]
	:	:	:	:	:
	100	X[5,100,1]	X[5,100,2]	...	X[5,100,4097]

Masing-masing Sinyal pada Tabel 3.1 akan dilakukan dekomposisi sinyal dengan metode DWT sebanyak  $N$  level yang akan digunakan sehingga akan dihasilkan  $N + 1$  sub-sinyal, dengan  $N$  menunjukkan level DWT yang digunakan. Berikut rincian jumlah variabel yang didapatkan pada masing-masing level:

**Tabel 3.2** Jumlah Sub-sinyal Tiap Level

Level	Jumlah Sub-sinyal
4	5
5	6
6	7

Masing-masing sub-sinyal akan diekstrak sebanyak 4 fitur yaitu *energy*, *variance*, *entropy* dan *line length*. Selanjutnya keempat fitur tersebut akan digunakan sebagai variabel dalam tahapan



klasifikasi dengan menggunakan metode SVM. Berikut merupakan variabel yang digunakan dengan menggunakan DWT level 4

**Tabel 3.3** Variabel Tahapan Klasifikasi DWT Level 4

<b>Variabel</b>	<b>Keterangan</b>	<b>Skala Data</b>
$Y$	Kelas	Nominal
$X_1$	<i>Energy</i> Sub-sinyal A4	Rasio
$X_2$	<i>Energy</i> Sub-sinyal D4	Rasio
$X_3$	<i>Energy</i> Sub-sinyal D3	Rasio
$X_4$	<i>Energy</i> Sub-sinyal D2	Rasio
$X_5$	<i>Energy</i> Sub-sinyal D1	Rasio
$X_6$	<i>Variance</i> Sub-sinyal A4	Rasio
$X_7$	<i>Variance</i> Sub-sinyal D4	Rasio
$X_8$	<i>Variance</i> Sub-sinyal D3	Rasio
$X_9$	<i>Variance</i> Sub-sinyal D2	Rasio
$X_{10}$	<i>Variance</i> Sub-sinyal D1	Rasio
$X_{11}$	<i>Entropy</i> Sub-sinyal A4	Rasio
$X_{12}$	<i>Entropy</i> Sub-sinyal D4	Rasio
$X_{13}$	<i>Entropy</i> Sub-sinyal D3	Rasio
$X_{14}$	<i>Entropy</i> Sub-sinyal D2	Rasio
$X_{15}$	<i>Entropy</i> Sub-sinyal D1	Rasio
$X_{16}$	<i>Line length</i> Sub-sinyal A4	Rasio
$X_{17}$	<i>Line length</i> Sub-sinyal D4	Rasio
$X_{18}$	<i>Line length</i> Sub-sinyal D3	Rasio
$X_{19}$	<i>Line length</i> Sub-sinyal D2	Rasio
$X_{20}$	<i>Line length</i> Sub-sinyal D1	Rasio

Struktur data yang digunakan dalam tahapan klasifikasi dengan DWT level 4 ditunjukkan pada Tabel 3.4.

**Tabel 3.4** Struktur Data Tahapan Klasifikasi

Sinyal	$X_1$	$X_2$		$X_{19}$	$X_{20}$	$Y$
1	$X_{1,1}$	$X_{2,1}$	...	$X_{19,1}$	$X_{20,1}$	$Y_1$
2	$X_{1,2}$	$X_{2,2}$	...	$X_{19,2}$	$X_{20,2}$	$Y_2$
3	$X_{1,2}$	$X_{2,3}$	...	$X_{19,3}$	$X_{20,3}$	$Y_3$
$\vdots$	$\vdots$	$\vdots$			$\vdots$	$\vdots$
500	$X_{1,500}$	$X_{2,500}$	...	$X_{19,500}$	$X_{20,500}$	$Y_{500}$

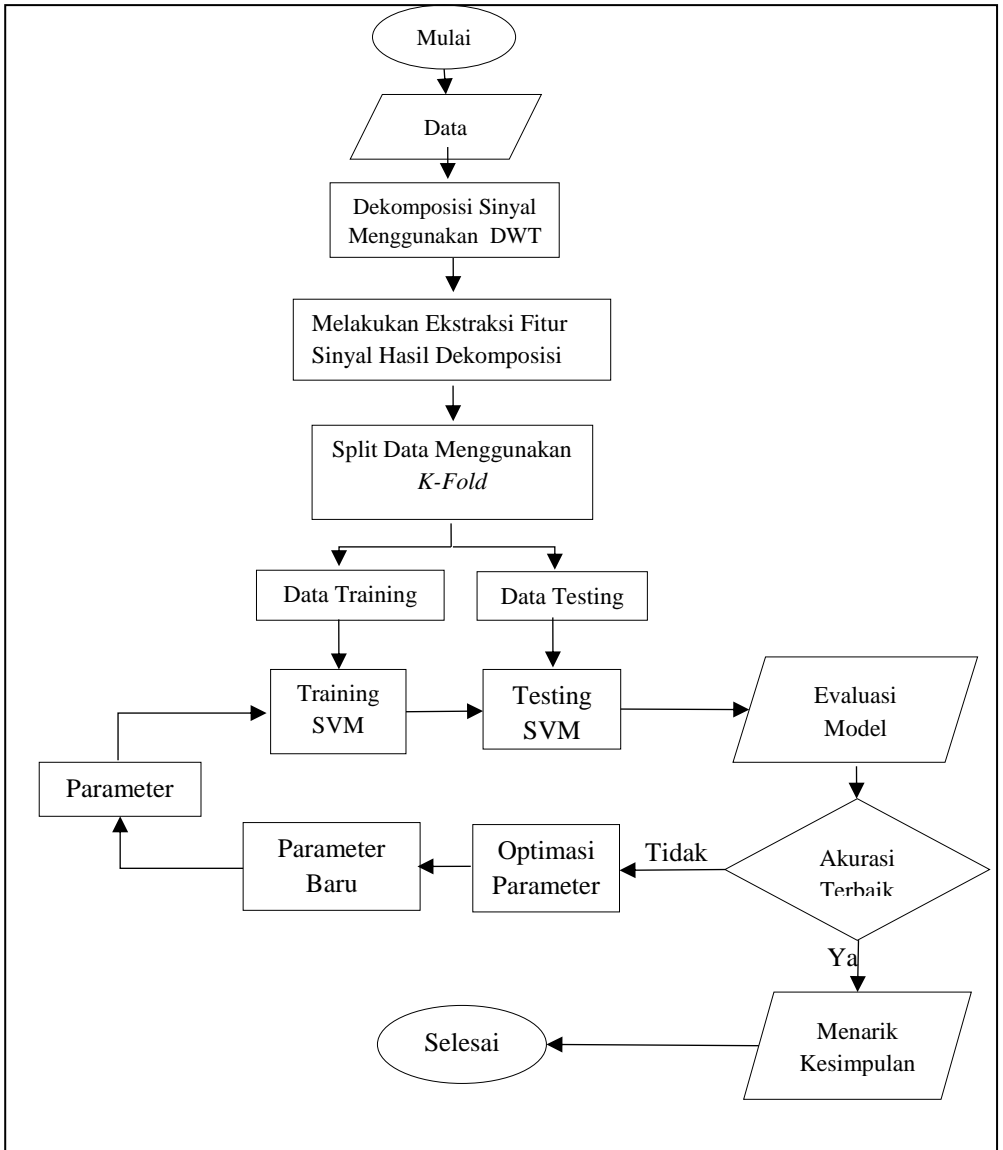
### 3.4 Langkah Analisis

Berikut ini adalah langkah analisis yang digunakan dalam klasifikasi epilepsi menggunakan data EEG.

1. Mengumpulkan data sekunder hasil rekaman EEG
2. Melakukan dekomposisi sinyal dengan metode DWT level 4, level 5 dan level 6 dengan langkah analisis sebagai berikut.
  - a. Memfilter atau mendekomposisi sinyal menjadi dua bagian yaitu frekuensi tinggi (H) dan rendah (L).
  - b. Melakukan proses *sub-band* pada masing masing filter, sehingga akan dihasilkan 4 buah *sub-band* pada dekomposisi level 1 (HH, HL-LH,LL)
  - c. Melakukan kembali langkah 1 dan 2 pada masing-masing *sub-band* sebanyak level yang diinginkan.
  - d. Menerapkan langkah 1 dan 2 pada masing-masing hasil dekomposisi level 3, level 4 dan level 5 sehingga akan dihasilkan dekomposisi level 4, 5 dan 6.
3. Melakukan ekstraksi fitur *energy*, *variance*, *entropy* dan *line length* pada masing-masing sub-sinyal

4. Melakukan split data menggunakan *stratified 5-Cross validation*
5. Melakukan klasifikasi menggunakan metode SVM dengan optimasi paraneter *grid search*
  - a. Menentukan parameter *Kernel* yang digunakan.
  - b. Menentukan *range* nilai parameter  $C$  dan  $\gamma$
  - c. Melakukan klasifikasi SVM dengan kombinasi nilai parameter  $C$  dan  $\gamma$ serta parameter *kernel* yang telah ditentukan.
  - d. Menghitung akurasi klasifikasi
  - e. Mengulangi langkah 5b. sampai semua kombinasi parameter telah dilakukan
  - f. Menghitung kinerja klasifikasi.

Langkah-langkah analisis secara umum dapat digambarkan dalam diagram alir pada Gambar 3.1.



**Gambar 3.1** Diagram Alir Langkah Analisis Penelitian

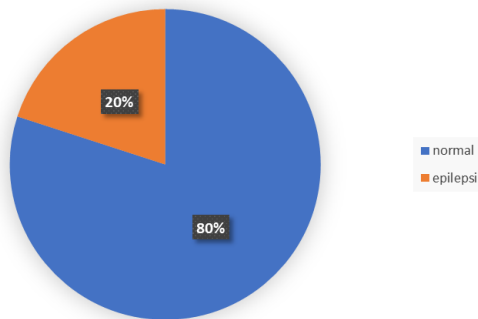
## BAB IV

### ANALISIS DAN PEMBAHASAN

Pada penelitian ini, analisis dan pemahasan mencakup empat tahap yaitu eksplorasi data, dekomposisi, ekstraksi fitur dan klasifikasi. Tahap eksplorasi data dilakukan untuk menunjukkan plot awal dari data. Tahap selanjutnya adalah *dekomposisi* dilakukan dengan DWT. Ekstraksi fitur yang dilakukan untuk mendapatkan fitur-fitur yang akan digunakan pada tahap klasifikasi. Tahapan klasifikasi dilakukan setelah didapatkan variabel-variabel dari tahap dekomposisi dan ekstraksi fitur.

#### 4.1 Eksplorasi Data Sinyal EEG

Eksplorasi data dilakukan untuk mengetahui karakteristik data secara visual. Karakteristik dapat ditunjukkan secara visual melalui beberapa *tools*. Pada penelitian ini akan dilakukan analisis eksplorasi data untuk melihat karakteristik dari data awal dan variabel respon dari data menggunakan *pie chart*.

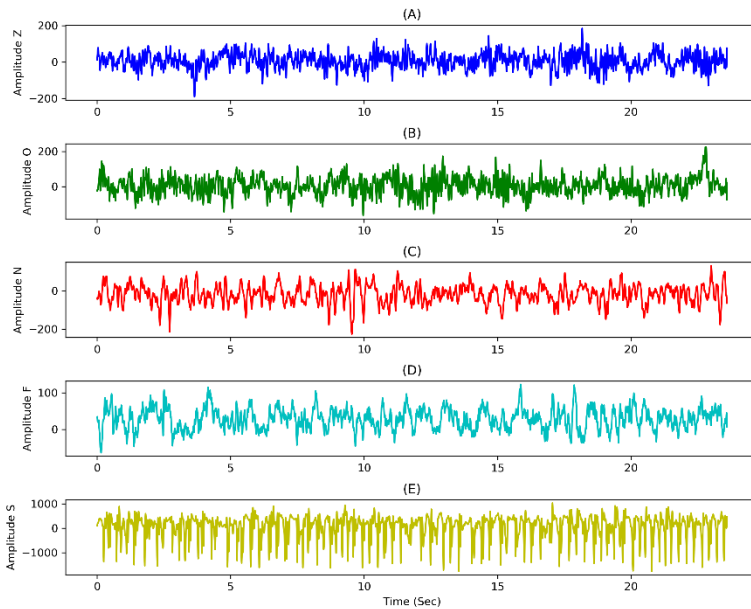


**Gambar 4.1** Perbandingan Kategori Data

Gambar 4.1 menunjukkan presentase dari data yang digunakan dalam penelitian ini. Terdapat 80% atau 400 data adalah pasien normal, sedangkan 20% sisanya merupakan pasien penderita epilepsi. Nilai 400 pengamatan terdiri dari 4 set data yaitu Z, N, O dan F. Keempat set data diamati dari pasien yang

tidak mengalami kejang dengan kondisi pengamatan yang berbeda. Sedangkan 20% atau 100 pengamatan lainnya adalah set data S yaitu pasien yang sedang diamati dari pasien yang sedang mengalami kejang.

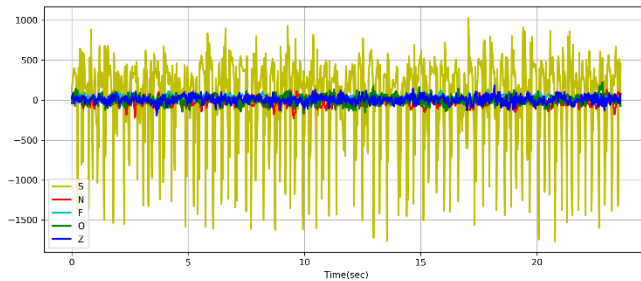
*Plot* dari masing-masing set data diperlukan untuk melihat karakteristik dari data sebelum dilakukan proses *filter*. *Plot* yang ditampilkan berikut merupakan lima set data yang mana dari masing masing set data hanya diambil satu dari seratus data yang ada dikarenakan keterbatasan tempat.



**Gambar 4.2** Plot Amplitudo Data Terhadap Waktu (a) Data Z, (b) Data O, (c) N, (d) F dan (e) S

Gambar 4.2 menunjukkan plot dari masing-masing set data. Plot tersebut menunjukkan batas-batas amplitudo yang berbeda-beda pada setiap set data. Hal ini menyebabkan tidak dapat diidentifikasi secara visual perbedaan dari masing-masing plot. Langkah selanjutnya akan dibentuk plot yang menggabungkan

kelima set data dalam satu frame untuk melihat lebih jelas perbedaan dari masing-masing set data.



**Gambar 4.3** Plot Gabungan 5 Set Data

Gambar 4.3 merupakan gabungan dari plot a,b,c,d dan e pada Gambar 4.2 dalam sebuah. Gambar 4.3 menunjukkan perbedaan yang sangat jelas dari set data S terhadap empat set data lainnya. Set data S merupakan set data pasien epilepsi yang direkam saat mengalami kejang sedangkan 4 set data lainnya merupakan set data yang direkam dalam keadaan tidak kejang. Data yang digunakan tersedia pada lampiran 1.

## 4.2 Dekomposisi Sinyal Menggunakan DWT

Dekomposisi data dilakukan sebelum ekstraksi fitur. Metode DWT akan digunakan untuk memproses lima set data hasil rekaman EEG. Proses ini akan mendekomposisi sinyal menjadi sub-sinyal rendah dan tinggi. Sinyal tinggi akan dieliminasi dan sinyal rendah akan didekomposisi lagi menjadi dua, proses tersebut merupakan proses dekomposisi level. Dekomposisi level 1 akan menghasilkan sub-sinyal A1, A2 dan D2. Level dekomposisi yang akan digunakan sebanyak tiga yaitu level 4, level 5 dan level 6. Pada proses DWT akan ditentukan fungsi *wavelet* yang akan digunakan. Fungsi *wavelet* yang akan digunakan pada penelitian ini adalah *Coiflets* orde 4 (*coif4*). Berikut merupakan proses DWT untuk masing-masing level yang digunakan :

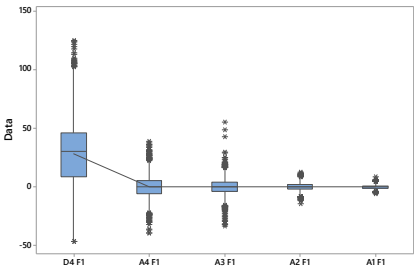
4.2.1 DWT Level 4

DWT level 4 akan mendekomposisi sebanyak 4 tahap. Proses ini akan menghasilkan 5 sub-sinyal pada setiap sinyal yang didekomposisi. Berikut sub-sinyal yang dihasilkan:

Tabel 4.1 Sub-Sinyal Level 4

	A4 F1	...	D4 F1	...	D1 Z100
1	34,12281	...	-1,65574	...	-2,08117
2	31,81413	...	-0,89998	...	5,23109
3	29,32116	...	-0,12235	...	-2,53283
4	26,65554	...	0,60965	...	-0,51497
5	23,86661	...	1,41789	...	-0,97566
⋮	⋮	⋮	⋮	⋮	⋮
4096	-17,7994	...	14,1183	...	4,370718
4097	-39,7844	...	9,96525	...	4,344681

Tabel 4.1 menunjukkan sub-sinyal yang dihasilkan dari proses DWT level 4. Berdasarkan Tabel dan penjelasan sebelumnya dapat diketahui terdapat 2500 sub-sinyal yang dihasilkan pada DWT level 4. Setelah didapatkan sub-sinyal dari proses DWT level 4 selanjutnya akan dibuat *boxplot* menggunakan 5 sub-sinyal yang dihasilkan dari salah satu sinyal. Berikut merupakan *boxplot* dari sinyal F1 setelah dilakukan proses DWT:



Gambar 4.4 Boxplot Level 4



Gambar 4.4 menunjukkan *boxplot* dari set data F1 setelah dilakukan dekomposisi DWT level 4. Gambar tersebut menunjukkan keempat sub-sinyal aproksimasi memiliki median yang sama hanya. Keempat sub-sinyal aproksimasi (A) tersebut memiliki perbedaan pada penyebaran data dimana sub-sinyal pada level dekomposisi tinggi memiliki penyebaran data yang lebih besar dibandingkan level dekomposisi kecil. Sub-sinyal detail (D) memiliki median yang lebih besar dibandingkan dengan sub-sinyal A serta penyebara data yang lebih besar.

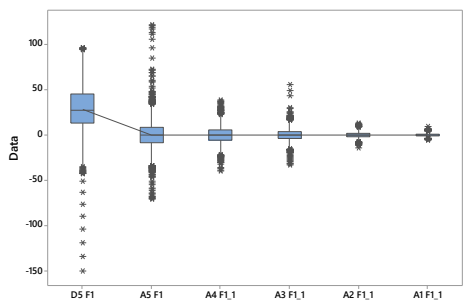
#### 4.2.2 DWT Level 5

DWT level 5 akan akan menghasilkan 6 sub-sinyal pada setiap sinyal yang didekomposisi. Berikut sub-sinyal yang dihasilkan:

**Tabel 4.2** Sub-Sinyal Level 5

	A5 F1	...	D5 F1	...	D1 Z100
1	11,894	...	22,227	...	-2,081
2	8,883	...	22,930	...	5,231
3	5,954	...	23,367	...	-2,532
4	3,124	...	23,531	...	-0,514
5	0,397	...	23,468	...	-0,975
⋮	⋮	⋮	⋮	⋮	⋮
4096	-17,799	...	116,500	...	4,370
4097	-39,784	...	110,407	...	4,344

Tabel 4.2 menunjukkan sub-sinyal yang dihasilkan dari proses DWT level 5. Berdasarkan Tabel dan penjelasan sebelumnya dapat diketahui terdapat 3000 sub-sinyal yang dihasilkan pada DWT level 5. Setelah didapatkan sub-sinyal dari proses DWT level 5 selanjutnya akan dibuat *boxplot* menggunakan 6 sub-sinyal yang dihasilkan dari salah satu sinyal. Sinyal yang digunakan pada *boxplot* ini adalah sinyal F1. Berikut merupakan *boxplot* dari sinyal F1 setelah dilakakukan proses DWT:



Gambar 4.5 *Boxplot* Level 5

Gambar 4.5 merupakan *boxplot* yang dibentuk dari set data F1 setelah dilakukan dekomposisi DWT level 5. Gambar tersebut menunjukkan informasi yang mirip pada Gambar 4.4, setiap sub-sinyal A memiliki median yang relatif sama dan sub-sinyal D memiliki median yang lebih besar dibandingkan keseluruhan sub-sinyal A. Perbedaannya adalah pada Gambar 4.5 sub-sinyal A5 memiliki *outlier* atas yang semakin menyebar dan melebihi outlier atas dari sub-sinyal D5 serta penyebaran outlier bawah dari D5 lebih besar dibandingkan pada D4 di Gambar 4.4.

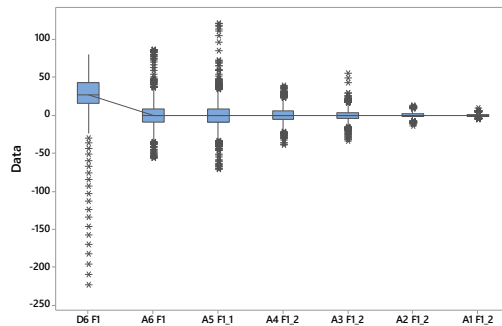
4.2.3 DWT Level 6

DWT level 6 akan akan menghasilkan 7 sub-sinyal pada setiap sinyal yang didekomposisi. Berikut sub-sinyal yang dihasilkan:

Tabel 4.3 Sub-Sinyal Level 6

	A6 F1	...	D6 F1	...	D1 Z100
1	12,757	...	-0,862	...	-2,081
2	11,615	...	-2,731	...	5,231
3	10,522	...	-4,568	...	-2,532
4	9,482	...	-6,358	...	-0,514
⋮	⋮	⋮	⋮	⋮	⋮
4096	-209,5	...	75,2007	...	4,370
4097	-223,599	...	73,406	...	4,344

Tabel 4.3 menunjukkan sub-sinyal yang dihasilkan dari proses DWT level 6. Berdasarkan Tabel dan penjelasan sebelumnya dapat diketahui terdapat 3500 sub-sinyal yang dihasilkan pada DWT level 5. Setelah didapatkan sub-sinyal dari proses DWT level 6 selanjutnya akan dibuat *boxplot* menggunakan 7 sub-sinyal yang dihasilkan dari salah satu sinyal. Sinyal yang digunakan pada *boxplot* ini adalah sinyal F1. Berikut merupakan *boxplot* dari sinyal F1 setelah dilakukan proses DWT:



**Gambar 4.6** *Boxplot* Level 6

Gambar 4.6 dapat disimpulkan pada setiap level sub-sinyal A cenderung memiliki median yang sama perbedaan sub-sinyal A pada setiap level terdapat pada persebaran data dan outlier. Begitu pula pada sub sinyal D memiliki median yang cenderung sama pada setiap levelnya perbedaan dengan sub-sinyal A terletak pada interkuartil box Sub-sinyal B cenderung tetap yang membesar hanya persebaran outliersnya. Hal ini ditunjukkan dengan semakin membesarnya skala gambar pada level yang lebih tinggi juga pada gambar terlihat outliers yang semakin panjang.

### 4.3 Ekstraksi Fitur

Ekstraksi fitur merupakan bagian penting dari proses pengolahan sinyal. Sinyal yang telah diperoleh pada proses sebelumnya akan diubah ke dalam bentuk yang diperlukan.

Penelitian ini akan menggunakan empat fitur dari masing-masing sinyal sebagai berikut:

#### 4.3.1 *Energy*

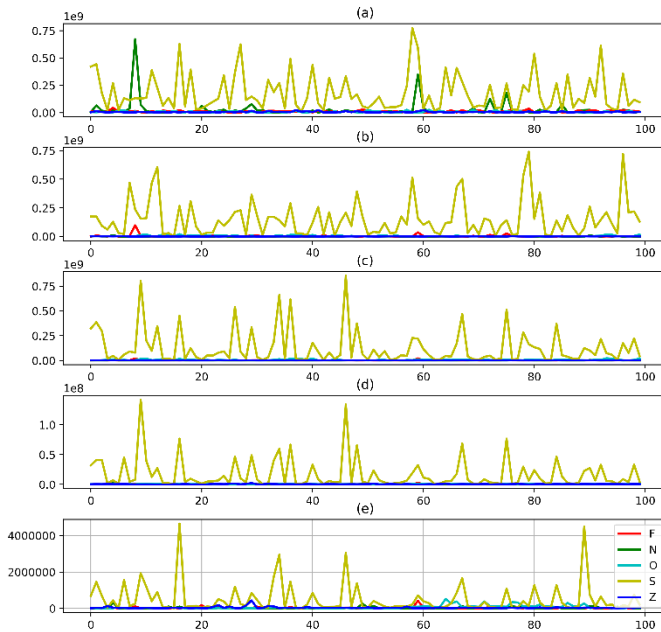
Energi dari sinyal didefenisikan sebagai jumlah kuadrat dari nilai modulus sampel. Penelitian mengambil fitur *energy* dari masing-masing sub-sinyal yang dihasil dari dekomposisi DWT. Sub-sinyal yang dihasilkan berbeda pada setiap level. Level 4 menghasilkan 5 sub-sinyal. Masing-masing sub-sinyal tersebut dilakukan ekstraksi fitur untuk mendapatkan fitur *energy* sehingga didapatkan data seperti pada Tabel 4.4 berikut:

**Tabel 4.4** Hasil Ekstraksi Fitur *Energy*

	<i>Energy</i>					Y
	A4	D4	D3	D2	D1	
1	6150648	328548,2	181045,8	38614,62	10906,77	0
2	66285950	8139940	2016515	243115,8	8221,49	0
3	18744496	4194421	1450006	179037,2	13579,07	0
4	9570030	1101784	222749,5	35305,62	9747,225	0
5	20343719	6679344	2588871	719956,1	42950,84	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
301	4,44E+08	1,75E+08	3,87E+08	40900606	1458990	1
302	1,8E+08	9308069 6	2,98E+08	40680479	686262,6	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	4078346	2163123	2287805	772216,9	20551,19	0
500	6527774	1641689	1987047	357141,6	44943,76	0

Data pada Tabel 4.4 merupakan hasil ekstraksi fitur *energy* yang dilakukan pada data hasil dekomposisi DWT level 4. Hasil ekstraksi fitur *energy* akan dibuat plot dari kelima set data Z, O, N, F dan S. Set data S merupakan sinyal otak pasien epilepsi yang sedang mengalami kejang, set data N dan F merupakan sinyal otak pasien epilepsi yang tidak mengalami kejang dimana set N diambil direkam pada pusat epilepsi dan F bukan dari pusat epilepsi. Set

data Z dan O merupakan sinyal otak dari relawan normal dimana set data O rekaman sinyal diambil saat mata terbuka dan set data Z diambil saat mata tertutup.



**Gambar 4.7** Plot Fitur *Energy* Kelima Set Data pada Tiap Sub-Sinyal (a) Sinyal A4, (b) Sinyal D4 dan (c) Sinyal D3, (d) Sinyal D2 dan (e) Sinyal D1.

Gambar 4.7 menunjukkan plot berwarna kuning merupakan *energy* dari set data S yang merupakan sinyal otak pasien epilepsi yang sedang kejang. Set data S terlihat memiliki plot yang lebih fluktuatif secara umum pada kelima sub-sinyal jika dibandingkan dengan keempat plot set data Z, O, N dan F. set data N dan F merupakan sinyal otak pasien epilepsi yang tidak mengalami kejang dimana set N direkam pada pusat epilepsi dan F bukan dari pusat epilepsi. Set data Z dan O merupakan sinyal otak dari relawan normal dimana set data O rekaman sinyal diambil saat mata terbuka dan set data Z diambil saat mata tertutup. Sub-sinyal A4

menunjukkan plot set data S cenderung memiliki nilai yang lebih besar hampir disetiap titik dibandingkan dengan keempat set data lainnya, pada beberapa titik plot set data N memiliki nilai yang lebih tinggi sedangkan pada sub-sinyal lainnya set data S memiliki nilai yang cenderung lebih tinggi dibandingkan keempat set data lainnya. Fluktuasi dari plot set data S lebih besar dibandingkan keempat set data pada kelima sub-sinyal A4, D4, D3, D2 dan D1. Terlihat pula pada Gambar 4.7 fluktuasi set data S pada sub-sinyal A4, D4 dan D3 lebih besar dibandingkan pada sub-sinyal D2 dan D1. Berdasarkan penjelasan tersebut dapat diketahui secara visual fitur *energy* dapat memberikan informasi dari perbedaan antara sinyal kejang dan non-kejang.

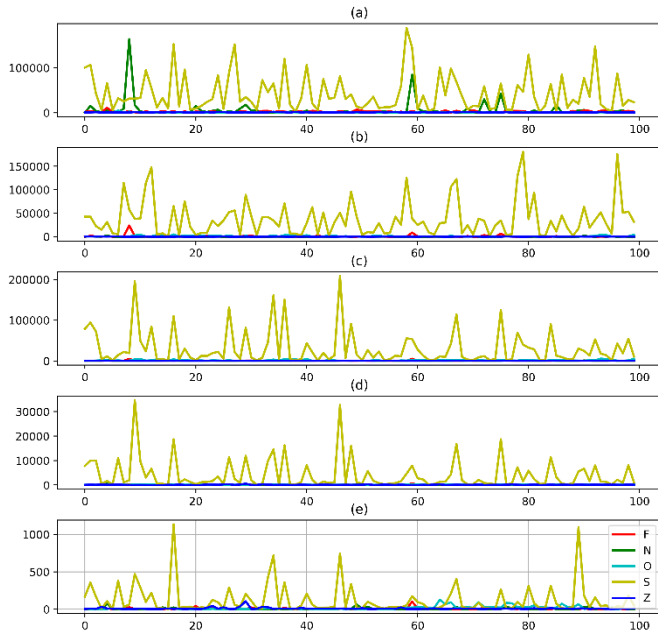
#### 4.3.2 Variance

Pengukuran *Variance* menunjukkan seberapa jauh persebaran dari sekumpulan data. *Variance* dari sinyal sebagai nilai persebaran dari data. Penelitian ini diambil fitur *variance* dari setiap sub-sinyal yang didapatkan dari hasil dekomposisi menggunakan DWT. Sub-sinyal yang dihasilkan berbeda pada setiap level. Level 4 menghasilkan 5 sub-sinyal. Masing-masing sub-sinyal tersebut diekstraksi fitur untuk mendapatkan fitur *variance* sehingga didapatkan data seperti pada Tabel 4.5.

**Tabel 4.5** Hasil Ekstraksi Fitur *Variance*

	<i>Variance</i>					Y
	A4	D4	D3	D2	D1	
1	686,396	8,017,259	4,417,906	9,422,795	2,661,486	0
2	15169,34	1,986,319	4,920,728	5,932,548	2,006,219	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
301	100612	42667,67	78787,5	7,803,717	1,678,484	1
302	106893,7	42591,01	94465,22	9,980,625	3,560,248	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	7,756,093	5,278,471	5,582,735	1,884,375	5,014,932	0
500	7,934,663	4,006,067	4,848,812	8,715,021	1,096,724	0

Data pada Tabel 4.5 merupakan hasil ekstraksi fitur *variance* yang dilakukan pada data hasil dekomposisi DWT level 4. Hasil ekstraksi fitur *variance* akan dibuat plot untuk menunjukkan sinyal kejang dan sinyal non kejang secara visual berdasarkan fitur *variance* pada masing-masing sub-sinyal.



**Gambar 4.8** Plot Fitur *Variance* Kelima Set Data pada tiap Sub-Sinyal (a) Sinyal A4, (b) Sinyal D4 dan (c) Sinyal D3, (d) Sinyal D2 dan (e) Sinyal D1.

Gambar 4.8 menunjukkan plot berwarna kuning merupakan *variance* dari set data S atau pasien epilepsi sedang mengalami kejang. Set data S terlihat memiliki plot yang lebih fluktuatif secara umum pada kelima sub-sinyal jika dibandingkan dengan keempat plot set data Z, O, N dan F. Set data N dan F merupakan sinyal otak pasien epilepsi yang tidak mengalami kejang dimana set N direkam pada pusat epilepsi dan F bukan dari pusat epilepsi. Set data Z dan O merupakan sinyal otak dari relawan normal dimana set data O rekaman sinyal diambil saat mata terbuka dan

set data Z diambil saat mata tertutup. Sub-sinyal A4 menunjukkan plot set data S cenderung memiliki nilai yang lebih besar hampir disetiap titik dibandingkan dengan keempat set data lainnya, pada beberapa titik plot set data N memiliki nilai yang lebih tinggi sedangkan pada sub-sinyal lainnya set data S memiliki nilai yang cenderung lebih tinggi dibandingkan keempat set data lainnya. Fluktuasi dari plot set data S lebih besar dibandingkan keempat set data pada kelima sub-sinyal A4, D4, D3, D2 dan D1. Terlihat pula pada gambar 4.8 fluktuasi set data S pada sub-sinyal A4 dan D4 lebih besar dibandingkan pada sub-sinyal D3, D2 dan D1. Berdasarkan penjelasan tersebut dapat diketahui secara visual fitur *variance* dapat memberikan informasi dari perbedaan antara sinyal kejang dan non-kejang.

### 4.3.3 Entropy

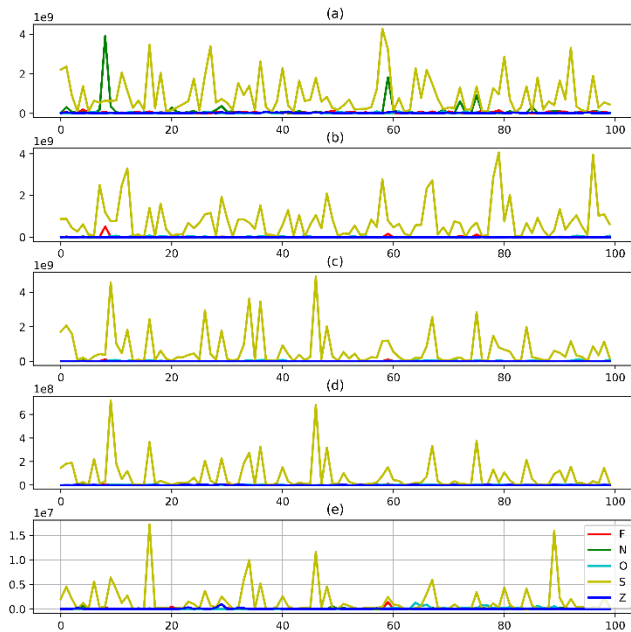
*Entropy* merupakan salah satu fitur yang sering digunakan dalam menganalisis sinyal biomedis. Penelitian ini diambil fitur *variance* dari setiap sub-sinyal yang didapatkan dari hasil dekomposisi menggunakan DWT. Sub-sinyal yang dihasilkan berbeda pada setiap level. Level 4 menghasilkan 5 sub-sinyal. Masing-masing sub-sinyal tersebut diekstraksi fitur untuk mendapatkan fitur *entropy* sehingga didapatkan data seperti pada Tabel 4.6.

**Tabel 4.6** Hasil Ekstraksi Fitur *Entropy*

	<i>Entropy</i>					Y
	A4	D4	D3	D2	D1	
1	21144400	748432,8	375630,8	50586,64	8,291,241	0
2	3,18E+08	34526356	7724240	713594,4	7,425,511	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
301	2,22E+09	8,74E+08	1,72E+09	1,47E+08	2090821	1
302	2,37E+09	8,85E+08	2,07E+09	1,82E+08	4572688	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	13338352	6733525	7079864	2041211	21657,29	0
500	22575674	4980854	6107382	829694,1	61444,35	0



Data pada Tabel 4.6 merupakan hasil ekstraksi fitur *entropy* yang dilakukan pada data hasil dekomposisi DWT level 4. Hasil ekstraksi fitur level 5 dan 6 dapat dilihat pada lampiran 15 dan 16. Hasil ekstraksi fitur *entropy* akan dibuat plot untuk menunjukkan sinyal kejang dan sinyal non kejang secara visual berdasarkan fitur *entropy* pada masing-masing sub-sinyal. Gambar 4.9 merupakan plot fitur *entropy* yang menunjukkan perbedaan antara kelima set data pada masing-masing sub-sinyal. Plot S merupakan set data kejang sedangkan set data Z, O, N dan F merupakan set data rekaman tidak mengalami kejang.



**Gambar 4.9** Plot Fitur *Entropy* Kelima Set Data pada tiap Sub-Sinyal (A) Sinyal A4, (B) Sinyal D4, (C) Sinyal D3, (D) Sinyal D2 dan (E) Sinyal D1

Gambar 4.9 plot dari fitur *entropy* kelima set data pada masing-masing sub-sinyal. Plot berwarna kuning merupakan *entropy* dari set data S atau pasien epilepsi yang sedang

mengalami kejang. Set data S terlihat memiliki plot yang lebih fluktuatif secara umum pada kelima sub-sinyal jika dibandingkan dengan keempat plot set data Z, O, N dan F. Set data N dan F merupakan sinyal otak pasien epilepsi yang tidak mengalami kejang dimana set N direkam pada pusat epilepsi dan F bukan dari pusat epilepsi. Set data Z dan O merupakan sinyal otak dari relawan normal dimana set data O rekaman sinyal diambil saat mata terbuka dan set data Z diambil saat mata tertutup. Sub-sinyal A4 menunjukkan plot set data S cenderung memiliki nilai yang lebih besar hampir disetiap titik dibandingkan dengan keempat set data lainnya, pada beberapa titik plot set data N memiliki nilai yang lebih tinggi sedangkan pada sub-sinyal lainnya set data S memiliki nilai yang cenderung lebih tinggi dibandingkan keempat set data lainnya. Fluktuasi dari plot set data S lebih besar dibandingkan keempat set data pada kelima sub-sinyal A4, D4, D3, D2 dan D1. Terlihat pula pada gambar 4.8 fluktuasi set data S yang paling besar adalah pada sub-sinyal A4 dan D4 lebih besar sedangkan fluktuasi set data S yang paling kecil terlihat pada sub-sinyal D1. Berdasarkan penjelasan tersebut dapat diketahui secara visual fitur *entropy* dapat memberikan informasi dari perbedaan antara sinyal kejang dan non-kejang.

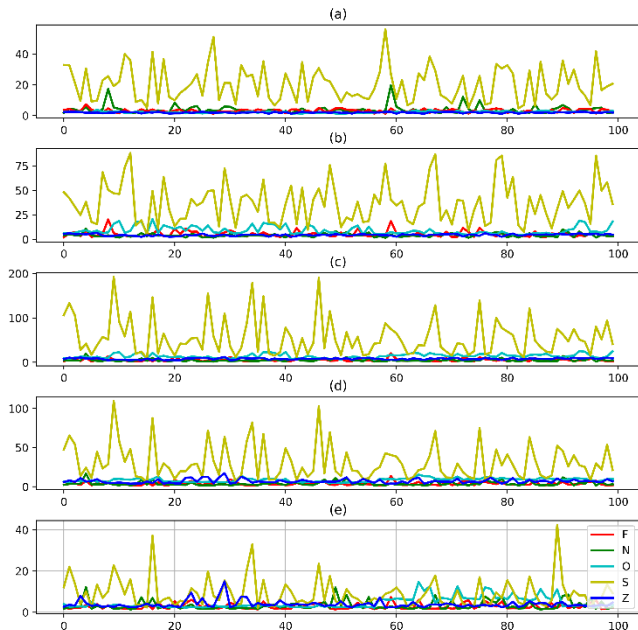
#### 4.3.4 *Line length*

*Line length* adalah ukuran kompleksitas sinyal atau *waveform fractal dimension*. (Esteller, Echauz, Tchong, T. Litt, & Pless, *Line Length: An Efficient Feature for Seizure Onset Detection*, 2001) menyatakan bahwa *line length* sensitif terhadap amplitudo dan variasi frekuensi sinyal. Penelitian ini diambil fitur *variance* dari setiap sub-sinyal yang didapatkan dari hasil dekomposisi menggunakan DWT. Sub-sinyal yang dihasilkan berbeda pada setiap level. Level 4 menghasilkan 5 sub-sinyal. Masing-masing sub-sinyal tersebut diekstraksi fitur untuk mendapatkan fitur *line length* sehingga didapatkan data seperti pada Tabel 4.7.

**Tabel 4.7** Hasil Ekstraksi Fitur *Line length*

	<i>Line length</i>					Y
	A4	D4	D3	D2	D1	
1	171,982	1,937,896	2,623,441	2,347,223	2,337,329	0
2	441,547	6,312,553	5,083,844	3,306,814	1,757,279	0
3	3.951884	6.951553	6.714888	4.028997	2.128517	0
4	2.688945	3.293269	2.977615	2.083406	2.180083	0
5	5.392671	8.68441	9.6877	7.0982	3.541071	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
301	3,294,557	4,825,872	1,065,582	4,739,936	1,218,587	1
302	3,272,712	4,205,808	1,332,813	6,538,698	221,119	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	2,302,188	5,533,134	103,176	9,512,975	3,001,291	0
500	1,916,857	4,680,621	8,696,101	6,636,259	4,755,733	0

Data pada Tabel 4.7 merupakan hasil ekstraksi fitur *line length* yang dilakukan pada data hasil dekomposisi DWT level 4. Hasil ekstraksi fitur level 5 dan 6 dapat dilihat pada lampiran 15 dan 16. Hasil ekstraksi fitur *line length* akan dibuat plot untuk menunjukkan sinyal kejang dan sinyal non kejang secara visual berdasarkan fitur *line length* pada masing-masing sub-sinyal. Gambar 4.8 merupakan plot fitur *line length* yang menunjukkan perbedaan antara kelima set data pada masing-masing sub-sinyal. Plot S merupakan set data kejang sedangkan set data Z, O, N dan F merupakan set data rekaman tidak mengalami kejang.



**Gambar 4.10** Plot Fitur *Line length* Kelima Set Data pada tiap Sub-Sinyal (A) Sinyal A4, (B) Sinyal D4, (C) Sinyal D3, (D) Sinyal D2 dan (E) Sinyal D1

Gambar 4.10 menunjukkan plot dari fitur *line length* pada masing-masing sub-sinyal level 4. Set data S terlihat memiliki plot yang lebih fluktuatif secara umum pada kelima sub-sinyal jika dibandingkan dengan keempat plot set data Z, O, N dan F. Set data N dan F merupakan sinyal otak pasien epilepsi yang tidak mengalami kejang dimana set N direkam pada pusat epilepsi dan F bukan dari pusat epilepsi. Set data Z dan O merupakan sinyal otak dari relawan normal dimana set data O rekaman sinyal diambil saat mata terbuka dan set data Z diambil saat mata tertutup. Sub-sinyal D1 menunjukkan plot set data S dan keempat set data lainnya cenderung memiliki nilai yang hampir sama pada beberapa titik berbeda dengan sub-sinyal lainnya set data S memiliki nilai yang cenderung lebih tinggi dibandingkan keempat set data lainnya. Plot

set data Z, O, N dan F pada sub-sinyal D1 lebih fluktuatif dibandingkan dengan sub-sinyal lainnya tetapi set data S masih memiliki fluktuasi yang lebih besar dibandingkan keempat set data lainnya pada sub-sinyal D1. Terlihat pula pada gambar 4.9 fluktuasi set data S yang sub-sinyal A4, D4, D3 dan D2 lebih besar dibandingkan pada sub-sinyal D1. Berdasarkan penjelasan tersebut dapat diketahui secara visual fitur *entropy* dapat memberikan informasi dari perbedaan antara sinyal kejang dan non-kejang.

#### 4.4 Support Vector Machine (SVM)

Klasifikasi merupakan tahap yang selanjutnya dilakukan setelah didapatkan fitur-fitur yang akan digunakan dalam tahap klasifikasi. Metode SVM akan digunakan sebagai *classifier* pada penelitian ini. SVM dapat menyelesaikan permasalahan klasifikasi pada data linier dan non-linier. Metode SVM yang akan digunakan pada penelitian ini adalah SVM dengan menggunakan kernel linier, polinomial dan RBF. *Grid search* akan digunakan sebagai metode untuk menentukan parameter yang menghasilkan model yang optimal. Metode pembagian data yang digunakan pada penelitian ini adalah *stratified 5-fold cross validation* pembagian data *training* dan *testing* dengan metode *stratified 5-fold cross validation*. Pada tahap awal data akan dibagi menjadi 5 bagian (*fold*) yang sama dengan proporsi yang sama antara data kejang dan non-kejang disetiap *fold*-nya. Sehingga dengan menggunakan *dataset* sebanyak 500 data dimana terdapat 100 data sinyal kejang dan 400 data sinyal non-kejang masing-masing *fold* akan beranggotakan 20 data sinyal kejang dan 80 data sinyal non-kejang.

##### 4.4.1 SVM Kernel Linear

Fungsi kernel linear digunakan terlebih dahulu karena memiliki fungsi kernel yang lebih sederhana tetapi menurut (Joachims, 1998) Metode ini sering memberikan hasil klasifikasi terbaik. Parameter  $C$  yang akan digunakan pada penelitian ditentukan berdasarkan akurasi terbaik yang dihasilkan pada  $C$  sebesar  $10^2$  sampai  $10^4$ . Parameter terbaik yang didapatkan pada

level 4 adalah  $C = 10$ . Setelah didapatkan parameter terbaik maka akan dibentuk fungsi *hyperplane* yang terbentuk untuk SVM *kernel* linear yang dapat dituliskan sebagai berikut:

$$f(\mathbf{x}) = \text{sign} \left( \sum_i^n a_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right)$$

Dimana  $\mathbf{x}$  adalah vektor dari variabel prediktor dan  $n$  adalah jumlah *support vector*. Setelah parameter  $C$  optimal diketahui kemudian dilakukan klasifikasi. Pembagian data dilakukan dengan metode *stratified 5-cross-validation*. Hasil pengukuran perfromansi klasifikasi masing-masing *fold*. Tabel 4.8 menunjukkan performa dari *5-fold* yang digunakan. Tebel performa tersebut didapatkan dari *confusion matrix* setiap *fold* yang disajikan pada lampiran 27. Berikut merupakan performa masing-masing *fold*:

**Tabel 4.8** Performa tiap *Fold* SVM Linear DWT Level 4

	Akurasi	Spesitifitas	Sensitivitas	AUC
1	0,94	0,9375	0,95	0,94375
2	0,99	1	0,95	0,975
3	1	1	1	1
4	1	1	1	1
5	0,99	1	0,95	0,975
Mean	0,984	0,9875	0,97	0,97875

Tabel 4.8 menunjukkan *fold* 3 dan 4 memberikan akurasi, sensitivitas, spesifisitas dan AUC yang terbaik sebesar 1. Dari hasil tersebut dapat diketahui pada *fold* tersebut sistem mampu mengklasifikasikan sinyal kejang dan non-kejang secara benar tanpa adanya misklasifikasi yang terjadi. Berdasarkan Tabel juga dapat diketahui rata-rata dari masing-masing performa klasifikasi. Rata-rata dari akurasi, spesitifitas, sensitivitas dan AUC yang dihasilkan masing-masing sebesar 98,4%, 98,75%, 97% dan 97,875%. Berikut merupakan *confusion matrix* dari SVM linear DWT level 4.

**Tabel 4.9** *Confusion Matrix* level 4

		Prediksi		
Aktual		Kejang	Non-Kejang	Total
	Kejang	97	3	100
	Non-Kejang	5	395	400
	Total	102	398	500

Tabel 4.9 menunjukkan prediksi dari SVM *kernel linear*. Tabel tersebut menunjukkan terdapat total misklasifikasi sebanyak 8 data sinyal dengan misklasifikasi sinyal non-kejang sebanyak 5 dan 3 misklasifikasi kejang dari 100 data sinyal kejang yang salah diklasifikasikan sebagai data sinyal non-kejang.

Parameter terbaik yang didapatkan pada DWT level 5 adalah  $C = 10$ . Selanjutnya dilakukan *split* data menggunakan metode *stratified 5-cross-validation*. Hasil pengukuran performansi klasifikasi masing-masing *fold*. Tabel 4.10 menunjukkan performa dari *5-fold* yang digunakan. Tabel performa tersebut didapatkan dari *confusion matrix* setiap *fold* yang disajikan pada lampiran 27. Berikut merupakan performa masing-masing *fold*:

**Tabel 4.10** Performa tiap *Fold* SVM Linear DWT Level 5

	Akurasi	Spesitifitas	Sensitivitas	AUC
1	0,98	0,9875	0,95	0,96875
2	0,94	1	0,7	0,85
3	0,98	1	0,9	0,95
4	1	1	1	1
5	0,99	1	0,95	0,975
Mean	0,978	0,9975	0,9	0,94875

Tabel 4.10 menunjukkan *fold* 4 memberikan akurasi, sensitivitas, spesifisitas dan AUC yang terbaik sebesar 1. Dari hasil tersebut dapat diketahui pada *fold* tersebut sistem mampu mengklasifikasikan sinyal kejang dan non-kejang secara benar tanpa adanya misklasifikasi yang terjadi. Berdasarkan Tabel juga dapat diketahui rata-rata dari masing-masing performa klasifikasi.

Rata-rata dari akurasi, spesitifitas, sensitivitas dan AUC yang dihasilkan masing-masing sebesar 97,8%, 99,75%, 90% dan 94,875%. Berikut merupakan *confusion matrix* dari SVM *linear* DWT level 5.

**Tabel 4.11** *Confusion Matrix* Level 5

Prediksi				
Aktual		Kejang	Non-Kejang	Total
	Kejang	90	10	100
	Non-Kejang	1	399	400
	Total	102	398	500

Tabel 4.11 menunjukkan prediksi dari SVM *kernel linear* pada DWT level 5. Tabel tersebut menunjukkan terdapat total misklasifikasi sebanyak 11 data sinyal dengan 1 misklasifikasi sinyal non-kejang dan 10 misklasifikasi kejang dari 100 data sinyal kejang yang salah diklasifikasikan sebagai data sinyal non-kejang.

Parameter terbaik yang didapatkan pada DWT level 6 adalah  $C = 10$ . Selanjutnya dilakukan *split* data menggunakan metode *stratified 5-cross-validation*. Hasil pengukuran perfromansi klasifikasi masing-masing *fold*. Tabel 4.12 menunjukkan performa dari *5-fold* yang digunakan. Tebel performa tersebut didapatkan dari *confusion matrix* setiap *fold* yang disajikan pada lampiran 28. Berikut merupakan performa masing-masing *fold*:

**Tabel 4.12** Hasil Performa SVM *Linear* tiap *Fold* DWT Level 6

	Akurasi	Spesitifitas	Sensitivitas	AUC
1	0,95	0,95	0,95	0,95
2	0,96	0,9875	0,85	0,91875
3	1	1	1	1
4	0,98	1	0,9	0,95
5	0,98	1	0,9	0,95
Mean	0,974	0,9875	0,92	0,95375

Tabel 4.12 menunjukkan *fold* 3 memberikan akurasi, sensitivitas, spesifisitas dan AUC yang terbaik sebesar 1. Dari hasil



tersebut dapat diketahui pada *fold* tersebut sistem mampu mengklasifikasikan sinyal kejang dan non-kejang secara benar tanpa adanya misklasifikasi yang terjadi. Berdasarkan Tabel juga dapat diketahui rata-rata dari masing-masing performa klasifikasi. Rata-rata dari akurasi, spesitifitas, sensitivitas dan AUC yang dihasilkan masing-masing sebesar 97,4%, 98,75%, 92% dan 95,375%. Berikut merupakan *confusion matrix* dari SVM *linear* DWT level 6.

**Tabel 4.13** *Confusion Matrix* Level 6

		Prediksi		
Aktual		Kejang	Non-Kejang	Total
	Kejang	95	5	100
	Non-Kejang	12	388	400
	Total	102	398	500

Tabel 4.13 menunjukkan total misklasifikasi sebanyak 17 data sinyal. Misklasifikasi sinyal non-kejang sebanyak 12 dan 5 dari 100 data sinyal kejang yang salah diklasifikasikan sebagai data sinyal non-kejang.

Rata-rata hasil performansi pada masing-masing level menunjukkan level 4 memberikan performa terbaik dengan akurasi terbaik sebesar 98,4% dan level 5 memberikan spesitivitas terbaik yaitu sebesar 99,75% yang berarti dari 400 data sinyal non-kejang hanya terdapat 1 misklasifikasi. Untuk sensitivitas terbaik diberikan pada level 4 yaitu sebesar 97% yang berarti dari 100 data sinyal kejang hanya terdapat 3 misklasifikasi. Serta performa AUC terbaik adalah pada DWT level 4 sebesar 97,875%.

#### 4.4.2 SVM *Kernel* RBF

Fungsi kernel RBF digunakan karena dapat memetakan hubungan tidak *linear*, RBF lebih robust terhadap outlier karena fungsi kernel RBF berada antara selang  $(-\infty, \infty)$ , sedangkan fungsi kernel lain memiliki rentang antara  $(-1, 1)$  (Hsu, C., & Lin, 2004). Parameter C dan  $\gamma$  yang digunakan akan ditentukan menggunakan prinsip *grid search* dengan C pada *range*  $10^{-2}$  -  $10^4$  dan  $\gamma$  pada *range*  $2^{-2}$  -  $2^4$ . Prinsip ini digunakan untuk mendapatkan parameter yang

menghasilkan akurasi terbaik pada masing-masing level DWT yang digunakan. Parameter terbaik yang didapatkan pada DWT level 4 adalah  $C = 10$  dan  $\gamma = 0,3$ . Fungsi *hyperplane* yang terbentuk untuk SVM *kernel* RBF dengan  $\gamma = 0,3$  dapat dituliskan sebagai berikut:

$$f(\mathbf{x}) = \text{sign} \left( \sum_i^n a_i y_i \exp(-0,3 \|\mathbf{x}_i - \mathbf{x}\|^2) + b \right)$$

Dimana  $\mathbf{x}$  adalah vektor dari variabel prediktor dan  $n$  adalah jumlah *support vector* dengan model terdapat pada lampiran 33. Pembagian data dilakukan menggunakan metode *stratified 5-cross-validation*. Hasil pengukuran perfomansi klasifikasi masing-masing *fold*. Tabel 4.14 menunjukkan performa dari *5-fold* yang digunakan. Tebel performa tersebut didapatkan dari *confusion matrix* setiap *fold* yang disajikan pada lampiran 29. Berikut merupakan performa masing-masing *fold*:

**Tabel 4.14** Hasil Performan SVM *Kernel* RBF tiap *Fold* pada DWT Level 4

	Akurasi	Spesitivitas	Sensitivitas	AUC
1	0,94	0,9375	0,95	0,94375
2	0,99	1	0,95	0,975
3	1	1	1	1
4	1	1	1	1
5	0,99	1	0,95	0,975
Mean	0,984	0,9875	0,97	0,97875

Tabel 4.14 menunjukkan *fold* 3 dan 4 memberikan akurasi, sensitivitas, spesifisitas dan AUC yang terbaik sebesar 1 sehingga dapat diketahui pada *fold* tersebut sistem mampu mengklasifikasikan sinyal kejang dan non-kejang secara benar tanpa adanya misklasifikasi yang terjadi. Berdasarkan Tabel juga dapat diketahui rata-rata dari masing-masing performa klasifikasi. Rata-rata dari akurasi, spesitifitas, sensitivitas dan AUC yang dihasilkan masing-masing sebesar 98,4%, 98,75%, 97% dan

97,875%. Berikut merupakan *confusion matrix* dari SVM *kernel* RBF pada DWT level 4.

**Tabel 4.15** *Confusion Matrix* Level 4

		Prediksi		
Aktual		Kejang	Non-Kejang	Total
	Kejang	97	3	100
	Non-Kejang	5	395	400
	Total	102	398	500

Tabel 4.15 menunjukkan prediksi dari SVM *kernel* RBF berdasarkan fungsi *hyperplane* yang dihasilkan. Pada Tabel dapat diketahui terdapat total misklasifikasi sebanyak 8 data sinyal. Misklasifikasi sinyal untuk sinyal non-kejang sebanyak 5 dari 400 data sinyal non-kejang sehingga didapatkan spesifisitas sebesar 98,75%. Misklasifikasi sinyal untuk sinyal kejang terdapat 3 data sinyal kejang yang salah diklasifikasikan sebagai data sinyal non-kejang sehingga didapatkan sensitivitas sebesar 97%.

SVM *kernel* RBF yang dilakukan pada DWT level 5 didapatkan Parameter terbaik adalah  $C = 100$  dan  $\gamma = 0,25$ . Berdasarkan parameter tersebut fungsi *hyperplane* yang terbentuk untuk SVM *kernel* RBF dengan  $\gamma = 2$  dapat dituliskan sebagai berikut:

$$f(\mathbf{x}) = \text{sign} \left( \sum_i^n a_i y_i \exp(-0,25 \|\mathbf{x}_i - \mathbf{x}\|^2) + b \right)$$

Pembagian data dilakukan dengan menggunakan metode *stratified 5-cross-validation*. Hasil pengukuran perfomansi klasifikasi masing-masing *fold*. Tabel 4.16 menunjukkan performa dari *5-fold* yang digunakan. Tebel performa tersebut didapatkan dari *confusion matrix* setiap *fold* yang disajikan pada lampiran 30. Berikut merupakan performa masing-masing *fold*:

**Tabel 4.16** Hasil Performa SVM *Kernel* RBF pada DWT Level 5

	Akuras	Spesitivitas	Sensitivitas	AUC
1	0,94	0,9375	0,95	0,94375
2	0,98	0,9875	0,95	0,96875
3	1	1	1	1
4	1	1	1	1
5	0,99	1	0,95	0,975
Mean	0,982	0,985	0,97	0,9775

Tabel 4.16 menunjukkan performa dari *5-fold* yang digunakan. Tabel menunjukkan *fold* 3 dan 4 memberikan akurasi, sensitivitas, spesifisitas dan AUC yang sempurna. Artinya pada *fold* tersebut sistem mampu mengklasifikasikan sinyal kejang dan non-kejang secara benar tanpa ada misklasifikasi. Rata-rata akurasi, spesifisitas, sensitivitas dan AUC yang dihasilkan masing-masing sebesar 98,2%, 98,5%, 97% dan 97,75%. Berikut merupakan *confusion matrix* dari SVM *kernel* RBF pada DWT level 5.

**Tabel 4.17** *Confusion Matrix* Level 5

		Prediksi		
Aktual		Kejang	Non-Kejang	Total
	Kejang	97	3	100
	Non-Kejang	6	394	400
	Total	102	398	500

Tabel 4.17 menunjukkan prediksi dari SVM *kernel* RBF. Tabel tersebut menunjukkan terdapat total misklasifikasi sebanyak 9 data sinyal dengan misklasifikasi sinyal non-kejang sebanyak 6 dan 3 misklasifikasi kejang dari 100 data sinyal kejang yang salah diklasifikasikan sebagai data sinyal non-kejang.

Parameter terbaik yang didapatkan pada DWT level 6 adalah  $C = 10$  dan  $\gamma = 2$ . Fungsi *hyperplane* yang terbentuk untuk SVM *kernel* RBF dengan  $\gamma = 2$  dapat dituliskan sebagai berikut:

$$f(x) = \text{sign} \left( \sum_i^n a_i y_i \exp(-2\|x_i - x\|^2) + b \right)$$

Selanjutnya dilakukan *split* data menggunakan metode *stratified 5-cross-validation*. Hasil pengukuran perfomansi klasifikasi masing-masing *fold* untuk SVM *kernel* RBF pada DWT level 5 ditunjukkan pada Tabel 4.18.

**Tabel 4.18** Hasil Performa SVM *Kernel* RBF pada DWT Level 6

	Akurasi	Spesitifitas	Sensitivitas	AUC
1	0,94	0,9375	0,95	0,94375
2	0,99	0,9875	1	0,99375
3	1	1	1	1
4	1	1	1	1
5	0,99	1	0,95	0,975
Mean	0,984	0,985	0,98	0,9825

Tabel 4.18 menunjukkan performa dari *5-fold* yang digunakan. Tabel menunjukkan *fold* 3 dan 4 memberikan akurasi, sensitivitas, spesifisitas dan AUC yang sempurna. Artinya pada *fold* tersebut sistem mampu mengklasifikasikan sinyal kejang dan non-kejang secara benar tanpa ada misklasifikasi. Rata-rata akurasi, Spesitifitas, sensitivitas dan AUC yang dihasilkan masing-masing sebesar 98,4%, 98,5%, 98% dan 98,25%. Berikut merupakan *confusion matrix* dari SVM *kernel* RBF pada DWT level 6.

**Tabel 4.19** *Confusion Matrix* Level 6

		Prediksi		
		Kejang	Non-Kejang	Total
Aktual	Kejang	98	2	100
	Non-Kejang	6	394	400
	Total	102	398	500

Tabel 4.19 menunjukkan total misklasifikasi sebanyak 8 data sinyal. Misklasifikasi sinyal non-kejang sebanyak 2 dan 6 dari 100 data sinyal kejang yang salah diklasifikasikan sebagai data sinyal non-kejang.

Rata-rata hasil performansi pada masing-masing level menunjukkan level 4 dan level 6 memberikan akurasi terbaik sebesar 98,4% dan level 4 memberikan spesitivitas terbaik yaitu sebesar 98,75% yang berarti dari 400 data sinyal non-kejang hanya terdapat 5 misklasifikasi. Untuk sensitivitas terbaik diberikan pada level 6 yaitu sebesar 98% yang berarti dari 100 data sinyal kejang hanya terdapat 3 misklasifikasi. Serta performa AUC terbaik adalah pada DWT level 4 sebesar 97,875%.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Hasil dekomposisi sinyal EEG dengan metode DWT pada level 4, 5 dan 6 masing-masing menghasilkan sub-sinyal sebanyak 5, 6 dan 7 sub sinyal. masing-masing subsinyal memberikan perbedaan yang signifikan antara sinyal kejang dan sinyal non-kejang. Kemudian dari hasil ekstraksi fitur sinyal dengan pada setiap sub-sinyal memberikan perbedaan yang signifikan antara sinyal kejang dan sinyal non-kejang.
2. Pengklasifikasian sinyal EEG dengan menggunakan metode SVM didapatkan performansi paling optimum pada DWT level 4 dan menggunakan *linear* dengan parameter  $C = 10$ , sedangkan untuk kernel RBF pada level DWT yang sama dengan parameter  $C = 10$  dan  $\gamma = 0,3$  dengan akurasi, spesitivitas, sensitivitas dan AUC masing-masing sebesar 98,4%, 98,75%, 97% dan 97,875%.

#### **5.2 Saran**

Berdasarkan kesimpulan yang diperoleh, dapat dirumuskan saran sebagai pertimbangan penelitian selanjutnya adalah sebagai berikut.

1. Penelitian selanjutnya sebaiknya menggunakan data real dari rekaman EEG rumah sakit untuk mengetahui apakah metode ini dapat digunakan pada kasus lain.
2. Perlu menambahkan level DWT yang digunakan untuk mengetahui pada level berapa yang menghasilkan fitur yang paling optimal.

*(Halaman ini sengaja dikosongkan)*



## DAFTAR PUSTAKA

- Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adeli, H. (2017). Deep convolutional neural network for the automated detection and diagnosis of seizures using EEG signals. *Computers in Biology and Medicine*, 1.
- Acharya, U. R., Sree, V., Chattopadhyay, S., Yu, W., & Alvin, A. (2011). Application of recurrence quantification analysis for the automated identification of epileptic EEG signals. *International Journal of Neural System*, 21, 199-211.
- Bekkar, Djema, M., & Alitouche, T. A. (2013). Evaluation Measures for Models Assesment over Imbalanced Datasets. *Journal of Information Engineering and Application*, 27-38.
- Burrus, C. S., Gopinath, R. A., & Gou, H. (1998). Introduction to Wavelets and Wavelet Transform. New Jersey: Prentice Hall.
- Cristianini, N., & Taylor, J. S. (2000). *An introduction to support vector machine: and other kernel-based learning methods*. New York: Cambrige University Press.
- Enas, A., Elamaran, V., Chandrasekar, M., Balaji, V. S., & Narasimhan, K. (2017). Automated diagnosis of epilepsy from EEG signals using ensemble learning approach. *Pattern Recognition Letters*, 1. Dipetik February 25, 2018, dari  
file:///D:/epilepsy,%20eeg,%20machine%20learning/1-s2.0-S0167865517301691-main.pdf

- Esteller, R., Echauz, J., & Tcheng, T. (2004). Comparison of *Line length* Feature Before and After Brain Electrical Simulation in Epileptic Patient. *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 4710-4713.
- Esteller, R., Echauz, J., Tcheng, T., T. Litt, B., & Pless, B. (2001). *Line length*: An Efficient Feature for Seizure Onset Detection. *23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1707-1710.
- Faust, O., Acharya, U. R., Adeli, H., & Adeli, A. (2015). Wevelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis. *Seizure*, 26, 56-64.
- Fitriyani, P. N., Tjandrasa, H., & Navastara, D. A. (2017). *Deteksi Penyakit Epilepsi Berdasarkan Data EEG Menggunakan Empirical Mode Decomposition dan Interval Type-2 FUZZY Logic System*. Surabaya, Jawa Timur, Indonesia: Institut Teknologi Sepuluh Nopember. Diambil kembali dari [http://repository.its.ac.id/41767/1/5113100081-Undergraduate\\_Theses.pdf](http://repository.its.ac.id/41767/1/5113100081-Undergraduate_Theses.pdf)
- Francisco, U. o. (2018, february 24). *University of California San Francisco Health*. Diambil kembali dari University of California San Francisco Health Web site: [https://www.ucsfhealth.org/conditions/neurological\\_disorders/](https://www.ucsfhealth.org/conditions/neurological_disorders/)
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing*. Prenice Hall.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Minig. Concept and Techniques*. Morgan Kaufmann.

- Hantoro, R. (2013). *Buku pintar keperawatan epilepsi*. Yogyakarta: Cakrawala ilmu.
- Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A Practical Guide to Support Vector Classification.
- Hsu, C., C., & Lin, C. (2004). *Guide to Support Vector Classification. Department of Computer Science and Information*. National Taiwan University.
- Joachims, T. (1998). *Text Categorization with Support Vector Machine: Learning with Many Relevant Features*.
- Kannathal, N., Choo, M. L., Acharya, U. R., & Sadasivan, P. K. (2005). Entropies for detection of epilepsy in EEG. *Computer Methods and Programs in Biomedicine*, 187-194.
- Kumar, A., & Kolekar, M. H. (2014). Machine Learning Approach for Epileptic Seizure Detection Using Wavelet Analysis of EEG Signals.
- Mallat, S. (1999). *A Wavelet Tour Of Signal Processing*. London.
- Mirzaei, A., Ayatollahi, A., Gifani, F., & Salehi, L. (2010). Spectral *Entropy* for Epileptic Seizures Detection.
- Nguyen, T., Khosravi, A., Creighton, D., & Nahavandi, S. (2015). EEG signal classification for BCI application by wavelets and interval type-2 fuzzy logic systems. *Expert system with application*, 42, 4370.
- Nugroho, A. S., Arief, B. W., & Dwi, H. (2003). Support Vector Machine Teori dan Aplikasinya dalam BioInformatika.
- Sandeep, K. S., Satchidananda, D., & Alok, K. J. (2017). EEG signal classification using PSO trained RBF neural

network for epilepsy identification. *Informatics in Medicine Unlocked*, 1.

Santoso, B. (2007). *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.

Silbernagl, S., & Lang, F. (2000). *Color atlas of pathophysiology*. New York: Thieme.

Starlen, K. J., Stel, V. S., Reitsma, J. B., Dekker, F. W., Zoccali, C., & Jager, J. K. (2009). diagnostic methods 1: sensitivity, specificity, and other measures of accuracy.

Tzallas, T. A., Tsipouras, G. M., & Fotiadis, I. D. (2007). The use of time-frequency distributions of epileptic seizure detection in EEG recordings. (hal. 3-6). Lyon: IEEE.

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.

WHO. (2018, maret 11). *WHO*. diakses dari world health organization web site: <http://www.who.int/mediacentre/factsheets/fs999/en/>

## LAMPIRAN

### Lampiran 1. Set Data EEG

No.	F1	F2	F3	F4	F5	...	Z98	Z99	Z100
1	34	60	26	-41	13	...	-51	56	-36
2	33	47	16	-42	6	...	-42	55	-71
3	28	38	13	-48	-1	...	-39	38	-120
4	22	29	12	-48	-13	...	-47	-5	-150
5	21	28	17	-48	-29	...	-51	-47	-160
6	22	30	16	-45	-42	...	-46	-72	-133
7	22	28	7	-44	-53	...	-29	-79	-96
8	19	30	-6	-36	-71	...	-7	-62	-56
9	22	36	-19	-42	-86	...	9	-39	-28
10	18	51	-25	-43	-94	...	10	-21	-39
11	22	53	-25	-39	-97	...	-18	-13	-56
12	25	62	-27	-37	-93	...	-27	-23	-76
13	15	65	-30	-33	-83	...	-51	-23	-93
14	5	69	-36	-32	-88	...	-59	-22	-87
15	0	72	-49	-22	-86	...	-53	-1	-95
16	-8	79	-50	-28	-99	...	-29	18	-96
17	-10	82	-64	-32	-118	...	4	12	-98
18	-19	79	-78	-29	-131	...	20	-29	-67
19	-25	75	-88	-37	-144	...	51	-74	-38
20	48	51	24	-65	-44	...	-58	-5	-72
21	51	57	45	-69	-32	...	-92	-11	-93
22	51	59	65	-67	-18	...	-97	-22	-92
...	...	...	...	...	...	...	...	...	...
4094	39	149	114	-30	161	...	-2	-32	3
4095	41	126	99	-23	129	...	0	-4	-13
4096	7	42	-130	-13	1	...	-49	69	30

## Lampiran 2. Syntax DWT Level 4

```
import numpy as np
import pywt
import pywt.data
mode = pywt.Modes.smooth
w = pywt.Wavelet('coif4')
x= range(500)
y=range(4)
ca=dict()
cd=dict()
rec_a=dict()
rec_d=dict()

for i in range(0,500):
    a = df.iloc[:,i]
    ca[i] = []
    cd[i] = []
    for j in y:
        (a, d) = pywt.dwt(a, w, mode)
        ca[i].append(a)
        cd[i].append(d)

    rec_a[i] = []
    rec_d[i] = []

    for j, dwt4f in enumerate(ca[i]):
        dwt4f_list = [dwt4f, None] + [None]
* j

rec_a[i].append(pd.DataFrame(pywt.waverec(dw
t4f_list, w)))

    for j, dwt4f in enumerate(cd[i]):
        dwt4f_list = [None, dwt4f] + [None]
* j
```

**Lampiran 2. Syntax DWT Level 4 (lanjutan)**

```

rec_d[i].append(pd.DataFrame(pywt.waverec(dw
t4f_list, w)))
dwt4 = pd.DataFrame()
for i in range (500):
    dwt4=pd.concat([dwt4,rec_a[i][3]],
axis=1)
for i in range (500):
    dwt4=pd.concat([dwt4,rec_d[i][3]],
axis=1)
for i in range (500):
    dwt4=pd.concat([dwt4,rec_d[i][2]],
axis=1)
for i in range (500):
    dwt4=pd.concat([dwt4,rec_d[i][1]],
axis=1)
for i in range (500):
    dwt4=pd.concat([dwt4,rec_d[i][0]],
axis=1)
from itertools import product
L = ['A4 F','A4 N', 'A4 O','A4 S', 'A4
Z','D4 F','D4 N','D4 O','D4 S','D4 Z','D3
F','D3 N','D3 O','D3 S','D3 Z','D2 F','D2
N','D2 O','D2 S','D2 Z','D1 F','D1 N','D1
O','D1 S','D1 Z']
k = 100
L2 = range(1, k+1)
dwt4.columns= [x+ str(y) for x,y in
list(product(L,L2))]
coef=dwt4.dropna(axis=0)
coef.to_csv('Coef5.csv')

```

**Lampiran 3. Syntax DWT Level 5**

```

import numpy as np
import pywt
import pywt.data
mode = pywt.Modes.smooth
w = pywt.Wavelet('coif4')
x= range(500)
y=range(5)
ca=dict()
cd=dict()
rec_a=dict()
rec_d=dict()

for i in range(0,500):
    d = df.iloc[:,i]
    cd[i] = []
    ca[i] = []
    for j in y:
        (d, a) = pywt.dwt(d, w, mode)
        cd[i].append(d)
        ca[i].append(a)

    rec_d[i] = []
    rec_a[i] = []

    for j, coeff in enumerate(cd[i]):
        coeff_list = [coeff, None] + [None]
* j

rec_d[i].append(pd.DataFrame(pywt.waverec(coeff_list, w)))

    for j, coeff in enumerate(ca[i]):
        coeff_list = [None, coeff] + [None]
* j

```



### Lampiran 3. Syntax DWT Level 5 (lanjutan)

```

rec_a[i].append(pd.DataFrame(pywt.waverec(coeff_list, w)))
dwt5 = pd.DataFrame()
for i in range (500):
    dwt5=pd.concat([dwt5,rec_d[i][4]],
axis=1)
for i in range (500):
    dwt5=pd.concat([dwt5,rec_a[i][4]],
axis=1)
for i in range (500):
    dwt5=pd.concat([dwt5,rec_a[i][3]],
axis=1)
for i in range (500):
    dwt5=pd.concat([dwt5,rec_a[i][2]],
axis=1)
for i in range (500):
    dwt5=pd.concat([dwt5,rec_a[i][1]],
axis=1)
for i in range (500):
    dwt5=pd.concat([dwt5,rec_a[i][0]],
axis=1)
from itertools import product
L = ['D5 F','D5 N', 'D5 O','D5 S', 'D5
Z','A5 F','A5 N','A5 O','A5 S','A5 Z','A4
F','A4 N','A4 O','A4 S','A4 Z','A3 F','A3
N','A3 O','A3 S','A3 Z','A2 F','A2 N','A2
O','A2 S','A2 Z','A1 F','A1 N','A1 O','A1
S','A1 Z']
k = 100
L2 = range(1, k+1)
dwt5.columns= [x+ str(y) for x,y in
list(product(L,L2))]
coef=dwt5.dropna(axis=0)

```

**Lampiran 4. Syntax DWT Level 6**

```

import pywt
import pywt.data
mode = pywt.Modes.smooth
w = pywt.Wavelet('coif4')
x= range(500)
y=range(6)
ca=dict()
cd=dict()
rec_a=dict()
rec_d=dict()

for i in range(0,500):
    d = df.iloc[:,i]
    cd[i] = []
    ca[i] = []
    for j in y:
        (d, a) = pywt.dwt(d, w, mode)
        cd[i].append(d)
        ca[i].append(a)

    rec_d[i] = []
    rec_a[i] = []

    for j, coeff in enumerate(cd[i]):
        coeff_list = [coeff, None] + [None]
* j

rec_d[i].append(pd.DataFrame(pywt.waverec(coeff_list, w)))

    for j, coeff in enumerate(ca[i]):
        coeff_list = [None, coeff] + [None]
* j
rec_a[i].append(pd.DataFrame(pywt.waverec(coeff_list, w)))

```

**Lampiran 4. Syntax DWT Level 6 (lanjutan)**

```

dwt6 = pd.DataFrame()
for i in range (500):
    dwt6=pd.concat([dwt6,rec_d[i][5]],
axis=1)
for i in range (500):
    dwt6=pd.concat([dwt6,rec_a[i][5]],
axis=1)
for i in range (500):
    dwt6=pd.concat([dwt6,rec_a[i][4]],
axis=1)
for i in range (500):
    dwt6=pd.concat([dwt6,rec_a[i][3]],
axis=1)
for i in range (500):
    dwt6=pd.concat([dwt6,rec_a[i][2]],
axis=1)
for i in range (500):
    dwt6=pd.concat([dwt6,rec_a[i][1]],
axis=1)
for i in range (500):
    dwt6=pd.concat([dwt6,rec_a[i][0]],
axis=1)
from itertools import product
L = ['A6 F','A6 N', 'A6 O','A6 S', 'A6
Z','D6 F','D6 N','D6 O','D6 S','D6 Z','D5
F','D5 N','D5 O','D5 S','D5 Z','D4 F','D4
N','D4 O','D4 S','D4 Z','D3 F','D3 N','D3
O','D3 S','D3 Z','D2 F','D2 N','D2 O','D2
S','D2 Z','D1 F','D1 N','D1 O','D1 S','D1
Z']
k = 100
L2 = range(1, k+1)
dwt6.columns= [x+ str(y) for x,y in
list(product(L,L2))]
coef=dwt6.dropna(axis=0)

```

**Lampiran 5. Hasil DWT Level 4**

<b>No.</b>	<b>A4 F1</b>	<b>A4 F2</b>	<b>...</b>	<b>A4 Z100</b>	<b>...</b>	<b>D4 F2</b>	<b>...</b>	<b>D1 Z100</b>
0	34.12281	65.71237	...	-22.188	...	-3.99479	...	-2.08117
1	31.81413	60.6254	...	-36.79	...	-8.79154	...	5.231095
2	29.32116	56.61163	...	-48.6238	...	-13.5893	...	-2.53283
3	26.65554	53.52899	...	-58.0531	...	-18.2027	...	-0.51497
4	23.86661	51.29324	...	-65.3026	...	-21.8589	...	-0.97566
5	20.99921	49.83954	...	-70.5459	...	-23.8718	...	2.673603
6	18.11716	49.01888	...	-74.1768	...	-23.4296	...	-1.13107
7	15.27395	48.69484	...	-76.5538	...	-19.8903	...	-1.25393
8	12.42838	48.80493	...	-77.82	...	-13.9608	...	2.605515
9	9.519334	49.28948	...	-78.1061	...	-6.72459	...	-2.79219
10	6.516748	50.12715	...	-77.451	...	1.314252	...	1.691356
11	3.382554	51.29605	...	-75.8926	...	9.450068	...	0.516288
12	0.101723	52.66833	...	-73.752	...	16.74806	...	-3.48682
13	-3.32177	54.11052	...	-71.3693	...	22.47573	...	5.177719
14	-6.92919	55.50989	...	-69.0174	...	25.43487	...	-3.89123
15	-10.7403	56.7678	...	-66.9387	...	24.67927	...	2.036173
16	-14.688	57.89563	...	-65.1116	...	20.91039	...	-1.6773
17	-18.6919	58.94025	...	-63.424	...	15.21797	...	2.137515
18	-22.638	59.91126	...	-61.8785	...	8.316074	...	-3.28401
19	-26.4114	60.83418	...	-60.434	...	1.117302	...	3.247455
:	:	:	:	:	:	:	:	:
4095	1.070499	166.9566	...	59.05764	...	-140.598	...	-12.6202
4096	-17.7994	108.3702	...	83.99356	...	-147.231	...	4.370718
4097	-39.7844	41.57018	...	111.9337	...	-142.045	...	4.344681

**Lampiran 6.** Hasil DWT Level 5

<b>No.</b>	<b>A5 F1</b>	<b>A5 F2</b>	<b>...</b>	<b>D4 F1</b>	<b>...</b>	<b>D4 Z99</b>	<b>...</b>	<b>D1 Z100</b>
0	11.89485	110.8691	...	-1.65574	...	10.35878	...	-2.08117
1	8.883906	103.9288	...	-0.89998	...	0.46852	...	5.231095
2	5.954165	97.2586	...	-0.12235	...	-10.3585	...	-2.53283
3	3.124091	90.88488	...	0.609655	...	-21.1925	...	-0.51497
4	0.397916	84.81808	...	1.417895	...	-30.5558	...	-0.97566
5	-2.22043	79.06464	...	2.424749	...	-37.2582	...	2.673603
6	-4.72751	73.64286	...	3.713966	...	-39.5305	...	-1.13107
7	-7.11867	68.56865	...	5.353239	...	-35.9781	...	-1.25393
8	-9.36961	63.85398	...	7.163834	...	-27.8217	...	2.605515
9	-11.4512	59.51197	...	8.878563	...	-16.9547	...	-2.79219
10	-13.3459	55.54323	...	10.4638	...	-4.38931	...	1.691356
11	-15.0345	51.94881	...	11.82939	...	8.495456	...	0.516288
12	-16.4942	48.74983	...	12.61823	...	20.20332	...	-3.48682
13	-17.7069	45.96581	...	12.52234	...	29.61958	...	5.177719
14	-18.6473	43.61386	...	11.1593	...	34.7729	...	-3.89123
15	-19.2962	41.70609	...	8.23657	...	34.11107	...	2.036173
16	-19.6654	40.22239	...	4.140537	...	28.75734	...	-1.6773
17	-19.7742	39.13205	...	-0.55826	...	20.46935	...	2.137515
18	-19.6408	38.41637	...	-5.56279	...	10.33462	...	-3.28401
19	-19.287	38.05057	...	-10.4796	...	-0.21332	...	3.247455
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
4095	-119.03	-171.693	...	16.77295	...	-46.4316	...	-12.6202
4096	-134.3	-215.433	...	14.11833	...	-41.2396	...	4.370718
4097	-150.192	-260.77	...	9.965253	...	-32.28	...	4.344681

**Lampiran 7. Hasil DWT Level 6**

<b>No.</b>	<b>A6 F1</b>	<b>A6 F2</b>	<b>...</b>	<b>D6 F1</b>	<b>...</b>	<b>D6 Z99</b>	<b>...</b>	<b>D1 Z100</b>
0	12.75713	150.4658	...	-0.86228	...	0.679647	...	-2.08117
1	11.61535	142.7499	...	-2.73144	...	-1.15844	...	5.231095
2	10.52229	135.2239	...	-4.56813	...	-2.96913	...	-2.53283
3	9.482156	127.8959	...	-6.35806	...	-4.73561	...	-0.51497
4	8.496543	120.7685	...	-8.09863	...	-6.45363	...	-0.97566
5	7.566734	113.844	...	-9.78716	...	-8.11945	...	2.673603
6	6.694808	107.1256	...	-11.4223	...	-9.72891	...	-1.13107
7	5.882792	100.6167	...	-13.0015	...	-11.2771	...	-1.25393
8	5.134343	94.32492	...	-14.504	...	-12.7432	...	2.605515
9	4.453641	88.25881	...	-15.9048	...	-14.1027	...	-2.79219
10	3.842888	82.42296	...	-17.1888	...	-15.3414	...	1.691356
11	3.304466	76.82235	...	-18.339	...	-16.4433	...	0.516288
12	2.842265	71.46467	...	-19.3365	...	-17.3878	...	-3.48682
13	2.459572	66.35625	...	-20.1665	...	-18.1581	...	5.177719
14	2.160323	61.50497	...	-20.8076	...	-18.7321	...	-3.89123
15	1.947453	56.91661	...	-21.2436	...	-19.0929	...	2.036173
16	1.818759	52.58629	...	-21.4841	...	-19.2512	...	-1.6773
17	1.770629	48.50619	...	-21.5449	...	-19.2248	...	2.137515
18	1.799932	44.66946	...	-21.4407	...	-19.0285	...	-3.28401
19	1.902823	41.06775	...	-21.1898	...	-18.6806	...	3.247455
:	:	:	:	:	:	:	:	:
4095	-195.908	-393.56	...	76.87822	...	-176.517	...	-12.6202
4096	-209.5	-427.152	...	75.20071	...	-168.751	...	4.370718
4097	-223.599	-462.036	...	73.40663	...	-160.735	...	4.344681

**Lampiran 8. Syntax Fitur *Energy* Level 4**

```

import pandas as pd
import numpy as np

energies = []

a = coef.as_matrix()
b=dict()
for column in range(0,2500):
    b[i]=coef.iloc[:,column].isnull().sum()
    energies.append(sum(a[0:a.shape[0]-
b[i]:,column] ** 2))

energy = pd.DataFrame(energies,
columns=['Energy'])
energy.to_csv('Energy4.csv')

```

**Lampiran 9. Syntax Fitur *Energy* Level 5**

```

import pandas as pd
import numpy as np

energies = []
a = coef.as_matrix()
b=dict()
for column in range(0,3000):
    b[i]=coef.iloc[:,column].isnull().sum()
    energies.append(sum(a[0:a.shape[0]-
b[i]:,column] ** 2))

energy = pd.DataFrame(energies,
columns=['Energy'])
energy.head()
energy.to_csv('Energy5.csv')

```

**Lampiran 10. Syntax Fitur *Energy* Level 6**

```

import numpy as np
import matplotlib.pyplot as plt

import pywt
import pywt.data
import pandas as pd
import numpy as np

energies = []

a = coef.as_matrix()
b=dict()
for column in range(3500):
    b[i]=coef.iloc[:,column].isnull().sum()
    energies.append(sum(a[0:a.shape[0]-
b[i]:,column] ** 2))

energy = pd.DataFrame(energies,
columns=['Energy'])
energy.head()
energy.to_csv('Energy6.csv')

```

**Lampiran 11. Data Fitur *Energy* Level 5**

	<i>Energy</i>						Y
	A5	D5	D4	D3	D2	D1	
<b>1</b>	5496454	968895.7	328548.2	181045.8	38614.62	10906.77	0
<b>2</b>	61049066	7976589	8139940	2016515	243115.8	8221.49	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>301</b>	1.68E+08	2.94E+08	1.75E+08	3.23E+08	31979632	687842.7	1
<b>302</b>	1.82E+08	2.61E+08	1.75E+08	3.87E+08	40900606	1458990	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>499</b>	3849139	1740760	2163123	2287805	772216.9	20551.19	0
<b>500</b>	6046096	1269913	1641689	1987047	357141.6	44943.76	0



**Lampiran 12. Data Fitur *Energy* Level 6**

	<i>Energy</i>							Y
	A6	D6	D5	D4	D3	D2	D1	
<b>1</b>	5180346	937595.6	968895.7	328548.2	181045.8	38614.62	10906.77	0
<b>2</b>	47774697	17656632	7976589	8139940	2016515	243115.8	8221.49	0
⋮	⋮		⋮	⋮	⋮	⋮	⋮	⋮
<b>301</b>	1.05E+08	1.66E+08	2.94E+08	1.75E+08	3.23E+08	31979632	687842.7	1
<b>302</b>	66683368	1.17E+08	2.61E+08	1.75E+08	3.87E+08	40900606	1458990	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>499</b>	5095602	2651750	1740760	2163123	2287805	772216.9	20551.19	0
<b>500</b>	6860907	2026358	1269913	1641689	1987047	357141.6	44943.76	0

**Lampiran 13. Syntax Fitur *Variance* Level 5**

```
import pandas as pd
import numpy as np

Variance = []

# a same as d
a = coef.as_matrix()
b=dict()
for column in range(0,3000):

b[i]=coef.iloc[:,column].isnull().sum()
    Variance.append(np.var(a[0:a.shape[0]-
b[i],column]))

Variance = pd.DataFrame(Variance,
columns=['Variance'])
Variance.head()
Variance.to_csv('Variance5.csv')
```

**Lampiran 14. Syntax Fitur *Variance* Level 6**

```

import pandas as pd
import numpy as np

Variance = []

# a same as d
a = coef.as_matrix()
b=dict()
for column in range(0,3500):

b[i]=coef.iloc[:,column].isnull().sum()
    Variance.append(np.var(a[0:a.shape[0]-
b[i],column]))

Variance = pd.DataFrame(Variance,
columns=['Variance'])
Variance.head()
Variance.to_csv('Variance6.csv')

```

**Lampiran 15. Data Fitur *Variance* Level 5**

	<i>Variance</i>						Y
	A5	D5	D4	D3	D2	D1	
<b>1</b>	532.36	236.42	80.17	44.18	9.42	2.66	0
<b>2</b>	13911.43	1946.36	1986.32	492.07	59.33	2.01	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>301</b>	38779.73	71849.88	42667.67	78787.50	7803.72	167.85	1
<b>302</b>	43113.24	63752.89	42591.01	94465.22	9980.63	356.02	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>499</b>	724.72	424.75	527.85	558.27	188.44	5.01	0
<b>500</b>	679.06	309.88	400.61	484.88	87.15	10.97	0

**Lampiran 16. Data Fitur *Variance* Level 6**

	<i>Variance</i>							Y
	A6	D6	D5	D4	D3	D2	D1	
1	465.76	228.76	236.42	80.17	44.18	9.42	2.66	0
2	10738.69	4307.44	1946.36	1986.32	492.07	59.33	2.01	0
⋮	⋮		⋮	⋮	⋮	⋮	⋮	⋮
301	22940.59	40445.12	71849.88	42667.67	78787.50	7803.72	167.85	1
302	14857.45	28485.59	63752.89	42591.01	94465.22	9980.63	356.02	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	1052.94	646.37	424.75	527.85	558.27	188.44	5.01	0
500	890.43	494.43	309.88	400.61	484.88	87.15	10.97	0

**Lampiran 17. Syntax Fitur *Entropy* Level 5**

```
import pandas as pd
import numpy as np
entropy=[]
b=dict()
result=dict()
entr=dict()
for i in range (0,3000):
    result[i] = coef.iloc[:,i]
    entropy.append(sum([(xi **
2)*np.log10(xi**2) for xi in result[i]]))
entropy=pd.DataFrame([entropy])
entropy=np.transpose(entropy)
entropy.columns=['entropy']
entropy.head()
entropy.to_csv('entropy5.csv')
```

**Lampiran 18. Syntax Fitur *Entropy* Level 6**

```
import pandas as pd
import numpy as np
entropy=[]
b=dict()
```

```

result=dict()
entr=dict()
for i in range (0,3500):
    result[i] = coef.iloc[:,i]
    entropy.append(sum([(xi **
2)*np.log10(xi**2) for xi in result[i]]))
entropy=pd.DataFrame([entropy])
entropy=np.transpose(entropy)
entropy.columns=['entropy']
entropy.to_csv('entropy6.csv')

```

### Lampiran 19. Data Fitur *Entropy* Level 5

	<i>Entropy</i>						Y
	A5	D5	D4	D3	D2	D1	
<b>1</b>	1.85E+07	2.82E+06	7.48E+05	3.76E+05	5.06E+04	8.29E+03	0
<b>2</b>	2.85E+08	3.29E+07	3.45E+07	7.72E+06	7.14E+05	7.43E+03	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>301</b>	8.49E+08	1.52E+09	8.74E+08	1.72E+09	1.47E+08	2.09E+06	1
<b>302</b>	9.32E+08	1.34E+09	8.85E+08	2.07E+09	1.82E+08	4.57E+06	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>499</b>	1.33E+07	5.79E+06	6.73E+06	7.08E+06	2.04E+06	2.17E+04	0
<b>500</b>	2.08E+07	3.73E+06	4.98E+06	6.11E+06	8.30E+05	6.14E+04	0

### Lampiran 20. Data Fitur *Entropy* Level 6

	<i>Entropy</i>							Y
	A6	D6	D5	D4	D3	D2	D1	
<b>1</b>	1.74E+07	2.67E+06	2.82E+06	7.48E+05	3.76E+05	5.06E+04	8.29E+03	0
<b>2</b>	2.14E+08	7.36E+07	3.29E+07	3.45E+07	7.72E+06	7.14E+05	7.43E+03	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>301</b>	6.02E+08	8.47E+08	1.52E+09	8.74E+08	1.72E+09	1.47E+08	2.09E+06	1
<b>302</b>	3.08E+08	5.73E+08	1.34E+09	8.85E+08	2.07E+09	1.82E+08	4.57E+06	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
<b>499</b>	2.04E+07	1.06E+07	5.79E+06	6.73E+06	7.08E+06	2.04E+06	2.17E+04	0
<b>500</b>	2.49E+07	7.28E+06	3.73E+06	4.98E+06	6.11E+06	8.30E+05	6.14E+04	0

**Lampiran 21. Syntax Fitur *Line length* Level 5**

```

import numpy as np

a=dict()
b=dict()
c=dict()
LLF=[]
for i in range (3000):
    a[i]=coef.iloc[0:(coef.shape[0]-
1),i].as_matrix()

b[i]=coef.iloc[1:(coef.shape[0]),i].as_mat
rix()
    c[i]=np.abs(b[i]-a[i])
    LLF.append(sum(c[i])/(coef.shape[0]-
1))
LLF = pd.DataFrame(LLF, columns=['LLF'])
LLF.head()
LLF.to_csv('LLF5.csv')

```

**Lampiran 22. Syntax Fitur *Line length* Level 6**

```

import numpy as np

a=dict()
b=dict()
c=dict()
LLF=[]
for i in range (3500):
    a[i]=coef.iloc[0:(coef.shape[0]-
1),i].as_matrix()

b[i]=coef.iloc[1:(coef.shape[0]),i].as_mat
rix()
    c[i]=np.abs(b[i]-a[i])

```

```
LLF.append(sum(c[i]) / (coef.shape[0] - 1))
LLF = pd.DataFrame(LLF, columns=['LLF'])
LLF.to_csv('LLF6.csv')
```

**Lampiran 23.** Data Fitur *Line length* Level 5

<i>Line length</i>							Y
	A5	D5	D4	D3	D2	D1	
1	0.95	1.54	1.94	2.62	2.35	2.34	0
2	3.44	3.51	6.31	5.08	3.31	1.76	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
301	11.70	29.99	48.26	106.56	47.40	12.19	1
302	9.31	30.43	42.06	133.28	65.39	22.11	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	1.02	2.19	5.53	10.32	9.51	3.00	0
500	0.99	1.83	4.68	8.70	6.64	4.76	0

**Lampiran 24.** Data Fitur *Line length* Level 6

<i>Line length</i>								Y
	A6	D6	D5	D4	D3	D2	D1	
1	0.54	0.79	1.54	1.94	2.62	2.35	2.34	0
2	2.46	2.98	3.51	6.31	5.08	3.31	1.76	0
⋮	⋮		⋮	⋮	⋮	⋮	⋮	⋮
301	3.27	11.39	29.99	48.26	106.56	47.40	12.19	1
302	3.03	8.85	30.43	42.06	133.28	65.39	22.11	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	0.69	0.90	2.19	5.53	10.32	9.51	3.00	0
500	0.67	0.88	1.83	4.68	8.70	6.64	4.76	0

**Lampiran 25** *Syntax SVM Kernel Linear*

```
from sklearn.svm import LinearSVC
svm=LinearSVC(C=100, tol=0.001,)
cv = StratifiedKFold(n_splits=5,
random_state=1)
```

```

#Setup arrays to store training and test
accuracies
n=5 #jumlah fold
from sklearn.model_selection import
StratifiedKFold
kf=StratifiedKFold(n_splits=n,
random_state=1000)

cm=[]
total=[]
ac=[]
se=[]
sp=[]

for train_index, test_index in
kf.split(X,y):
    X_train, X_test = X[train_index],
X[test_index]
    y_train, y_test = y[train_index],
y[test_index]
    svm.fit(X_train, y_train)
    y_pred = svm.predict(X_test)
    cm.append((confusion_matrix(y_test,
y_pred)).astype(float))
for j in range (n):
    total.append(sum(sum(cm[j])))

ac.append((cm[j][0,0]+cm[j][1,1])/total[j]
)
se.append(cm[j][0,0]/(cm[j][0,0]+cm[j][0,1]
))
sp.append(cm[j][1,1]/(cm[j][1,0]+cm[j][1,1]
))
akurasi=np.mean(ac)
spesifisiti=np.mean(se)
sensitiviti=np.mean(sp)

```

```
#performansi masing masing fold->csv
df_k3=pd.DataFrame()
test=dict()
for j in range (3):
    test[j]=[]
for i in range (n):
    test[0].append(ac[i])
    test[1].append(se[i])
    test[2].append(sp[i])
for i in range (3):

df_k3=pd.concat([df_k3,pd.DataFrame(test[i
])],axis=1)
df_k3.columns=['Akuras','Sensitivitas','Sp
esitifitas']
df_k3
```

### **Lampiran 26 Syntax SVM Kernel RBF**

```
from sklearn.svm import SVC
svm=SVC(C = 10, gamma=0.3,tol=0.001,kernel
= 'rbf', probability = True, class_weight=
None ,random_state=2)
cv = StratifiedKFold(n_splits=5,
random_state=1)

n=5
from sklearn.model_selection import
StratifiedKFold
kf=StratifiedKFold(n_splits=n,
random_state=1000)

cm=[]
total=[]
ac=[]
se=[]
sp=[]
```



```

a=dict()
auc=[]

for train_index, test_index in
kf.split(X,y):
    X_train, X_test = X[train_index],
X[test_index]
    y_train, y_test = y[train_index],
y[test_index]
    svm.fit(X_train, y_train)
    y_pred = svm.predict(X_test)
    cm.append((confusion_matrix(y_test,
y_pred)).astype(float))
for j in range (n):
    total.append(sum(sum(cm[j])))

ac.append((cm[j][0,0]+cm[j][1,1])/total[j]
)

se.append(cm[j][0,0]/(cm[j][0,0]+cm[j][0,1]
))

sp.append(cm[j][1,1]/(cm[j][1,0]+cm[j][1,1]
))
for j in range (n):
    auc.append((se[j]+sp[j])/2)
akurasi=np.mean(ac)
spesifisiti=np.mean(se)
sensitiviti=np.mean(sp)
#performansi masing masing fold->csv
df_k3=pd.DataFrame()
test=dict()
for j in range (4):
    test[j]=[]
for i in range (n):
    test[0].append(ac[i])

```

```
test[1].append(se[i])
test[2].append(sp[i])
test[3].append(auc[i])
for i in range (4):

df_k3=pd.concat([df_k3,pd.DataFrame(test[i
])],axis=1)
df_k3.columns=['Akuras','Sensitivitas','Sp
esitifitas','AUC']
df_k3
```

**Lampiran 27** *Confusion Matrix Kernel Linear Dwt Level 4 per Fold*

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	75	5	80
	Non-Kejang	1	19	20
	Total	76	24	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	1	19	20
	Total	81	19	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	0	20	20
	Total	80	20	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	0	20	20
	Total	80	20	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	1	19	20
	Total	81	19	100

**Lampiran 28** *Confusion Matrix Kernel RBF Dwt Level 4 per Fold*

	<b>Prediksi</b>		
<b>Aktual</b>	<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	75	5
	<b>Non-Kejang</b>	1	19
	<b>Total</b>	<b>76</b>	<b>24</b>

	<b>Prediksi</b>		
<b>Aktual</b>	<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	80	0
	<b>Non-Kejang</b>	1	19
	<b>Total</b>	<b>81</b>	<b>19</b>

	<b>Prediksi</b>		
<b>Aktual</b>	<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	80	0
	<b>Non-Kejang</b>	0	20
	<b>Total</b>	<b>80</b>	<b>20</b>

	<b>Prediksi</b>		
<b>Aktual</b>	<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	80	0
	<b>Non-Kejang</b>	0	20
	<b>Total</b>	<b>80</b>	<b>20</b>

	<b>Prediksi</b>		
<b>Aktual</b>	<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	80	0
	<b>Non-Kejang</b>	1	19
	<b>Total</b>	<b>81</b>	<b>19</b>

**Lampiran 29** *Confusion Matrix Kernel Linear Dwt Level 5 per Fold*

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	79	1	80
	Non-Kejang	1	19	20
	Total	80	20	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	2	18	20
	Total	82	18	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	2	18	20
	Total	82	18	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	3	17	20
	Total	83	17	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	1	19	20

	<b>Total</b>	<b>81</b>	<b>19</b>	<b>100</b>
--	--------------	-----------	-----------	------------

**Lampiran 30** *Confusion Matrix Kernel RBF Dwt Level 5 per Fold*

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	75	5	80
	Non-Kejang	1	19	20
	Total	76	24	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	79	1	80
	Non-Kejang	1	19	20
	Total	81	19	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	0	20	20
	Total	80	20	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	0	20	20
	Total	80	20	100

	Prediksi		
Aktua	Kejang	Non-Kejang	Total
	Kejang	80	0

	<b>Non-Kejang</b>	<b>1</b>	<b>19</b>	<b>20</b>
	<b>Total</b>	<b>81</b>	<b>19</b>	<b>100</b>

**Lampiran 31** *Confusion Matrix Kernel Linear Dwt Level 5 per Fold*

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	76	4	80
	Non-Kejang	1	19	20
	Total	79	23	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	79	1	80
	Non-Kejang	3	17	20
	Total	82	18	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	3	17	20
	Total	83	17	100

	Prediksi			
Aktual	Kejang	Non-Kejang	Total	
	Kejang	80	0	80
	Non-Kejang	1	19	20
	Total	81	19	100

	<b>Prediksi</b>		
<b>Aktual</b>	<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>

	<b>Kejang</b>	80	0	<b>80</b>
	<b>Non-Kejang</b>	2	18	<b>20</b>
	<b>Total</b>	<b>82</b>	<b>18</b>	<b>100</b>

**Lampiran 32** *Confusion Matrix Kernel RBF Dwt Level 5 per Fold*

	<b>Prediksi</b>			
<b>Aktual</b>		<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	75	5	<b>80</b>
	<b>Non-Kejang</b>	1	19	<b>20</b>
	<b>Total</b>	<b>76</b>	<b>24</b>	<b>100</b>

	<b>Prediksi</b>			
<b>Aktual</b>		<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	79	1	<b>80</b>
	<b>Non-Kejang</b>	0	20	<b>20</b>
	<b>Total</b>	<b>79</b>	<b>21</b>	<b>100</b>

	<b>Prediksi</b>			
<b>Aktual</b>		<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	80	0	<b>80</b>
	<b>Non-Kejang</b>	0	20	<b>20</b>
	<b>Total</b>	<b>80</b>	<b>20</b>	<b>100</b>

	<b>Prediksi</b>			
<b>Aktual</b>		<b>Kejang</b>	<b>Non-Kejang</b>	<b>Total</b>
	<b>Kejang</b>	80	0	<b>80</b>
	<b>Non-Kejang</b>	0	20	<b>20</b>
	<b>Total</b>	<b>80</b>	<b>20</b>	<b>100</b>

	<b>Prediksi</b>			
--	-----------------	--	--	--



Aktual	Kejang	Non-Kejang	Total
	Kejang	80	0
	Non-Kejang	1	19
	Total	81	19
			100

**Lampiran 33** Model SVM *Kernel* RBF pada DWT level 4

$$\begin{aligned}
& (1 \times 10 \times (-0,216x_1 - 0,264x_2 - 0,232x_3 + \dots - 0,5288x_{20})) \\
& + (1 \times 0,2236 \times (2,068x_1 + 2,088x_2 + 1,965x_3 + \dots + 2,798x_{20})) \\
& + (1 \times 0,1916 \times (3,727x_1 + 3,702x_2 + 7302x_3 + \dots + 3,9502x_{20})) \\
& + (-1 \times 10 \times (-0,2959x_1 - 0,3701x_2 - 0,3003x_3 + \dots - 0,355x_{20})) \\
& \quad \vdots \\
& + (1 \times 0,1295 \times (0,928x_1 + 0,9573x_2 + 0,844x_3 + \dots + 1,334x_{20})) \\
& \quad + 0,773
\end{aligned}$$

**Lampiran 34** Surat Pernyataan Data**SURAT PERNYATAAN**

Saya yang bertanda tangan di bawah ini, mahasiswa  
Departemen Statistika FMKSD ITS

Nama : Bayu Samudra

NRP : 062114 4000 0115

menyatakan bahwa data yang digunakan dalam Tugas Akhir ini merupakan data sekunder yang diambil dari penelitian/ buku/Tugas Akhir/Thesis/Publikasi/lainnya yaitu :

Sumber : Data dari website <http://epileptologie-bonn.de>

Keterangan : Data terdiri dari 5 set rekaman EEG (Z, O, N, F, S) setiap data terdiri dari 100 potong data sinyal masing-masing berdurasi 23,6 detik.

Surat pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka saya siap menerima sanksi sesuai aturan yang berlaku.

Surabaya, Juli 2018

Mengetahui,  
Pembimbing Tugas Akhir

Mahasiswa

Dr. Santi Wulan Purnami, M.Si

NIP. 19720923 199803 2 001

Bayu Samudra

NRP.062114 4000 0115

## BIODATA PENULIS



Penulis dengan nama lengkap Bayu Samudra dilahirkan di Kota Makassar pada 14 Januari 1997. Penulis menempuh pendidikan formal di SDN Unggulan Moginsidi 1 Makassar, SMPN 2 Makassar, dan SMAN 16 Makassar. Kemudian penulis diterima sebagai Mahasiswa Departemen Statistika ITS pada tahun 2014. Selama masa

perkuliahan, penulis pernah aktif dalam organisasi yang berfokus pada kejournalistikan dalam lingkup HIMASTA-ITS yaitu PERS HIMASTA-ITS staff PERS HIMASTA-ITS 2015/2016. Selama menjalani masa perkuliahan 4 tahun di Surabaya, penulis juga rutin melancong ke berbagai tempat yang ada di Pulau Jawa. hampir seluruh provinsi yang ada di Pulau Jawa telah dikunjungi oleh penulis. Apabila pembaca ingin memberi kritik dan saran serta diskusi lebih lanjut mengenai Tugas Akhir ini, dapat menghubungi penulis melalui email [bayu.samudra.009@gmail.com](mailto:bayu.samudra.009@gmail.com).